Informatik I: Einführung in die Programmierung

4. Funktionen: Aufrufe und Definitionen

UNI FREIBURG

Albert-Ludwigs-Universität Freiburg

Peter Thiemann

23. Oktober 2018

UNI FREIBURG

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Funktionsaufrufe

Funktionsaufrufe

UNI FREIBURG

- Funktionen sind Abbildungen von einem Definitionsbereich in einen Bildbereich.
- Eine Funktion erwartet Argumente aus dem Definitionsbereich und gibt einen Funktionswert (oder *Rückgabewert*) aus dem Bildbereich zurück.
- Eine Funktion kann Effekte haben, z.B.:
  - eine Ausgabe erzeugen,
  - eine Eingabe lesen,
  - uvam
- Viele Standardfunktionen sind in Python vordefiniert.

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Typen-Konversion

UNI FREIBURG

Die Funktionen `int`, `float`, `complex` und `str` können „passende“ Werte in den jeweiligen Typ umwandeln.

Python-Interpreter

```
>>> int(-2.6) # Umwandlung nach int durch Abschneiden.
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Typen-Konversion

UNI FREIBURG

Die Funktionen `int`, `float`, `complex` und `str` können „passende“ Werte in den jeweiligen Typ umwandeln.

Python-Interpreter

```
>>> int(-2.6) # Umwandlung nach int durch Abschneiden.
-2
>>>
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Typen-Konversion

UNI FREIBURG

Die Funktionen `int`, `float`, `complex` und `str` können „passende“ Werte in den jeweiligen Typ umwandeln.

Python-Interpreter

```
>>> int(-2.6) # Umwandlung nach int durch Abschneiden.
-2
>>> int('vier')
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Typen-Konversion

UNI FREIBURG

Die Funktionen `int`, `float`, `complex` und `str` können „passende“ Werte in den jeweiligen Typ umwandeln.

Python-Interpreter

```
>>> int(-2.6) # Umwandlung nach int durch Abschneiden.
-2
>>> int('vier')
 File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() ...
>>>
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Typen-Konversion

UNI FREIBURG

Die Funktionen `int`, `float`, `complex` und `str` können „passende“ Werte in den jeweiligen Typ umwandeln.

Python-Interpreter

```
>>> int(-2.6) # Umwandlung nach int durch Abschneiden.
-2
>>> int('vier')
 File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() ...
>>> complex('42')
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Typen-Konversion

UNI FREIBURG

Die Funktionen `int`, `float`, `complex` und `str` können „passende“ Werte in den jeweiligen Typ umwandeln.

Python-Interpreter

```
>>> int(-2.6) # Umwandlung nach int durch Abschneiden.
-2
>>> int('vier')
 File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() ...
>>> complex('42')
(42+0j)
>>>
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Typen-Konversion

UNI FREIBURG

Die Funktionen `int`, `float`, `complex` und `str` können „passende“ Werte in den jeweiligen Typ umwandeln.

Python-Interpreter

```
>>> int(-2.6) # Umwandlung nach int durch Abschneiden.
-2
>>> int('vier')
 File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() ...
>>> complex('42')
(42+0j)
>>> float(4)
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Typen-Konversion

UNI FREIBURG

Die Funktionen `int`, `float`, `complex` und `str` können „passende“ Werte in den jeweiligen Typ umwandeln.

Python-Interpreter

```
>>> int(-2.6) # Umwandlung nach int durch Abschneiden.
-2
>>> int('vier')
 File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() ...
>>> complex('42')
(42+0j)
>>> float(4)
4.0
>>>
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Typen-Konversion

UNI FREIBURG

Die Funktionen `int`, `float`, `complex` und `str` können „passende“ Werte in den jeweiligen Typ umwandeln.

Python-Interpreter

```
>>> int(-2.6) # Umwandlung nach int durch Abschneiden.
-2
>>> int('vier')
 File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() ...
>>> complex('42')
(42+0j)
>>> float(4)
4.0
>>> str(42)
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Typen-Konversion

UNI FREIBURG

Die Funktionen `int`, `float`, `complex` und `str` können „passende“ Werte in den jeweiligen Typ umwandeln.

Python-Interpreter

```
>>> int(-2.6) # Umwandlung nach int durch Abschneiden.
-2
>>> int('vier')
 File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() ...
>>> complex('42')
(42+0j)
>>> float(4)
4.0
>>> str(42)
'42'
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Ein-/Ausgabe

UNI FREIBURG

- Bekannt: `print` kann Werte ausgeben.
- Die Funktion `input` kann Strings einlesen.

Python-Interpreter

```
>>> input("Gib mir einen Keks: ")
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Ein-/Ausgabe

UNI FREIBURG

- Bekannt: `print` kann Werte ausgeben.
- Die Funktion `input` kann Strings einlesen.

Python-Interpreter

```
>>> input("Gib mir einen Keks: ")
Gib mir einen Keks:
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Ein-/Ausgabe

UNI FREIBURG

- Bekannt: `print` kann Werte ausgeben.
- Die Funktion `input` kann Strings einlesen.

Python-Interpreter

```
>>> input("Gib mir einen Keks: ")
Gib mir einen Keks: Keks
'Keks'
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Ein-/Ausgabe

UNI FREIBURG

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

- Bekannt: `print` kann Werte ausgeben.
- Die Funktion `input` kann Strings einlesen.

Python-Interpreter

```
>>> input("Gib mir einen Keks: ")
Gib mir einen Keks: Keks
'Keks'
>>> name = input("Wie heißt du? ")
Wie heißt du?
```

Standardfunktionen: Ein-/Ausgabe

UNI FREIBURG

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

- Bekannt: `print` kann Werte ausgeben.
- Die Funktion `input` kann Strings einlesen.

Python-Interpreter

```
>>> input("Gib mir einen Keks: ")
Gib mir einen Keks: Keks
'Keks'
>>> name = input("Wie heißt du? ")
Wie heißt du? Oskar
>>> print("Hallo,", name + "!")
Hallo, Oskar!
```

Standardfunktionen kombiniert: Eingabe von Zahlen

UNI FREIBURG

Da `input` nur Strings einliest, muss die Eingabe jeweils entsprechend konvertiert werden!

Python-Interpreter

```
>>> CM_PER_INCH = 2.54
>>> länge = input("Länge in cm: ")
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen kombiniert: Eingabe von Zahlen

UNI FREIBURG

Da `input` nur Strings einliest, muss die Eingabe jeweils entsprechend konvertiert werden!

Python-Interpreter

```
>>> CM_PER_INCH = 2.54
>>> länge = input("Länge in cm: ")
Länge in cm:
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen kombiniert: Eingabe von Zahlen

UNI FREIBURG

Da `input` nur Strings einliest, muss die Eingabe jeweils entsprechend konvertiert werden!

Python-Interpreter

```
>>> CM_PER_INCH = 2.54
>>> länge = input("Länge in cm: ")
Länge in cm: 195
>>> länge # ein String
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen kombiniert: Eingabe von Zahlen

UNI FREIBURG

Da `input` nur Strings einliest, muss die Eingabe jeweils entsprechend konvertiert werden!

Python-Interpreter

```
>>> CM_PER_INCH = 2.54
>>> länge = input("Länge in cm: ")
Länge in cm: 195
>>> länge # ein String
'195'
>>> länge_cm = float(länge)
>>> länge_inches = länge_cm / CM_PER_INCH
>>> print(länge + "cm", "=", str(länge_inches) + "in")
195cm = 76.77165354330708in
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Numerische Funktionen

UNI FREIBURG

- `abs` liefert den Absolutwert (auch bei `complex`)
- `round` rundet.

Python-Interpreter

```
>>> abs(-2)
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Numerische Funktionen

UNI FREIBURG

- `abs` liefert den Absolutwert (auch bei `complex`)
- `round` rundet.

Python-Interpreter

```
>>> abs(-2)
2
>>>
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Numerische Funktionen

UNI FREIBURG

- `abs` liefert den Absolutwert (auch bei `complex`)
- `round` rundet.

Python-Interpreter

```
>>> abs(-2)
2
>>> abs(1+1j)
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Numerische Funktionen

UNI FREIBURG

- `abs` liefert den Absolutwert (auch bei `complex`)
- `round` rundet.

Python-Interpreter

```
>>> abs(-2)
2
>>> abs(1+1j)
1.4142135623730951
>>>
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Numerische Funktionen

UNI FREIBURG

- `abs` liefert den Absolutwert (auch bei `complex`)
- `round` rundet.

Python-Interpreter

```
>>> abs(-2)
2
>>> abs(1+1j)
1.4142135623730951
>>> round(2.500001)
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Numerische Funktionen

UNI FREIBURG

- `abs` liefert den Absolutwert (auch bei `complex`)
- `round` rundet.

Python-Interpreter

```
>>> abs(-2)
2
>>> abs(1+1j)
1.4142135623730951
>>> round(2.500001)
3
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Zeichen-Konversion

UNI FREIBURG

Die Funktionen `chr` und `ord` wandeln Zahlen in Unicode-Zeichen um (und umgekehrt), wobei in Python Zeichen identisch mit einbuchstabigen Strings sind:

Python-Interpreter

```
>>> chr(42)
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Zeichen-Konversion

UNI FREIBURG

Die Funktionen `chr` und `ord` wandeln Zahlen in Unicode-Zeichen um (und umgekehrt), wobei in Python Zeichen identisch mit einbuchstabigen Strings sind:

Python-Interpreter

```
>>> chr(42)
'*'
>>>
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Zeichen-Konversion

UNI FREIBURG

Die Funktionen `chr` und `ord` wandeln Zahlen in Unicode-Zeichen um (und umgekehrt), wobei in Python Zeichen identisch mit einbuchstabigen Strings sind:

Python-Interpreter

```
>>> chr(42)
'*'
>>> chr(255)
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Zeichen-Konversion

UNI FREIBURG

Die Funktionen `chr` und `ord` wandeln Zahlen in Unicode-Zeichen um (und umgekehrt), wobei in Python Zeichen identisch mit einbuchstabigen Strings sind:

Python-Interpreter

```
>>> chr(42)
'*'
>>> chr(255)
'ÿ'
>>>
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Zeichen-Konversion

UNI FREIBURG

Die Funktionen `chr` und `ord` wandeln Zahlen in Unicode-Zeichen um (und umgekehrt), wobei in Python Zeichen identisch mit einbuchstabigen Strings sind:

Python-Interpreter

```
>>> chr(42)
'*'
>>> chr(255)
'ÿ'
>>> ord('*')
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Zeichen-Konversion

UNI FREIBURG

Die Funktionen `chr` und `ord` wandeln Zahlen in Unicode-Zeichen um (und umgekehrt), wobei in Python Zeichen identisch mit einbuchstabigen Strings sind:

Python-Interpreter

```
>>> chr(42)
'*'
>>> chr(255)
'ÿ'
>>> ord('*')
42
>>>
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Zeichen-Konversion

UNI FREIBURG

Die Funktionen `chr` und `ord` wandeln Zahlen in Unicode-Zeichen um (und umgekehrt), wobei in Python Zeichen identisch mit einbuchstabigen Strings sind:

Python-Interpreter

```
>>> chr(42)
'*'
>>> chr(255)
'ÿ'
>>> ord('*')
42
>>> ord('**')
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Standardfunktionen: Zeichen-Konversion

UNI FREIBURG

Die Funktionen `chr` und `ord` wandeln Zahlen in Unicode-Zeichen um (und umgekehrt), wobei in Python Zeichen identisch mit einbuchstabigen Strings sind:

Python-Interpreter

```
>>> chr(42)
'*'
>>> chr(255)
'ÿ'
>>> ord('*')
42
>>> ord('**')
Traceback (most recent call last): ...
TypeError: ord() expected a character, but string of
length 2 found
```


Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Kleiner Exkurs: Zeichenkodierung

UNI FREIBURG

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

- Computer können Berechnungen durchführen.

Kleiner Exkurs: Zeichenkodierung

UNI FREIBURG

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

- Computer können Berechnungen durchführen.
- Seit langem werden mit dem Computer auch Texte verarbeitet.

Kleiner Exkurs: Zeichenkodierung

UNI FREIBURG

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

- Computer können Berechnungen durchführen.
- Seit langem werden mit dem Computer auch Texte verarbeitet.
- Wie werden Texte im Computer dargestellt?

Kleiner Exkurs: Zeichenkodierung

UNI FREIBURG

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

- Computer können Berechnungen durchführen.
- Seit langem werden mit dem Computer auch Texte verarbeitet.
- Wie werden Texte im Computer dargestellt?
- Jeder Buchstabe entspricht einem Zahlenwert. Texte sind Sequenzen von solchen Codezahlen.

Kleiner Exkurs: Zeichenkodierung

UNI FREIBURG

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

- Computer können Berechnungen durchführen.
- Seit langem werden mit dem Computer auch Texte verarbeitet.
- Wie werden Texte im Computer dargestellt?
- Jeder Buchstabe entspricht einem Zahlenwert. Texte sind Sequenzen von solchen Codezahlen.
- Damit wird auch die **Textverabeitung** zu einer Berechnung.

ASCII

- Einer der ersten Zeichenkodes war ASCII (American Standard Code for Information Interchange) – entwickelt für Fernschreiber und Lochstreifen.

USASCII code chart

Row	Column 0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
10	LF	SUB	*	:	J	Z	j	z
11	VT	ESC	+	;	K	[	k	{
12	FF	FS	,	<	L	\	l	|
13	CR	GS	-	=	M	]	m	}
14	SO	RS	.	>	N	^	n	~
15	SI	US	/	?	O	_	o	DEL

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

ASCII

- Einer der ersten Zeichenkodes war ASCII (American Standard Code for Information Interchange) – entwickelt für Fernschreiber und Lochstreifen.

USASCII code chart

b7 b6 b5 →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b4 b3 b2 b1 ↓ Row / Column →	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

- Benötigt 7 Bits und enthält alle druckbaren Zeichen der englischen Sprache sowie nicht-druckbare Steuerzeichen (z.B. Zeilenwechsel).

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Erweitertes ASCII

- In anderen Sprachen wurden zusätzliche Zeichen benötigt.

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Erweitertes ASCII

- In anderen Sprachen wurden zusätzliche Zeichen benötigt.
- Da mittlerweile praktisch alle Rechner 8-Bit-Bytes als kleinste Speichereinheit nutzten, standden die höherwertigen Kodes (128–255) für Erweiterungen zur Verfügung.

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Erweitertes ASCII

UNI FREIBURG

- In anderen Sprachen wurden zusätzliche Zeichen benötigt.
- Da mittlerweile praktisch alle Rechner 8-Bit-Bytes als kleinste Speichereinheit nutzten, standen die höherwertigen Kodes (128–255) für Erweiterungen zur Verfügung.
- Diverse Erweiterungen, z.B. ISO-Latin-1 (mit Umlauten).

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Erweitertes ASCII

UNI FREIBURG

- In anderen Sprachen wurden zusätzliche Zeichen benötigt.
- Da mittlerweile praktisch alle Rechner 8-Bit-Bytes als kleinste Speichereinheit nutzten, standen die höherwertigen Kodes (128–255) für Erweiterungen zur Verfügung.
- Diverse Erweiterungen, z.B. ISO-Latin-1 (mit Umlauten).
- Auf dem IBM-PC gab es andere Erweiterungen, Windows-1252.

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Erweitertes ASCII

UNI FREIBURG

- In anderen Sprachen wurden zusätzliche Zeichen benötigt.
- Da mittlerweile praktisch alle Rechner 8-Bit-Bytes als kleinste Speichereinheit nutzten, standen die höherwertigen Kodes (128–255) für Erweiterungen zur Verfügung.
- Diverse Erweiterungen, z.B. ISO-Latin-1 (mit Umlauten).
- Auf dem IBM-PC gab es andere Erweiterungen, Windows-1252.
- Sprachen, die nicht auf dem lateinischen Alphabet basieren, haben große Probleme, ISO-2022-JP.

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Unicode

- Um für alle Sprachräume einen einheitlichen Zeichencode zu haben, wurde Unicode entwickelt (Version 1.0 im Jahr 1991).

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Unicode

UNI FREIBURG

- Um für alle Sprachräume einen einheitlichen Zeichencode zu haben, wurde Unicode entwickelt (Version 1.0 im Jahr 1991).
- Im Juni 2018 (Version 11.0.0) unterstützt Unicode 146 Schriften mit 137374 Codepoints, darunter 1212 Emojis.

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Unicode

UNI FREIBURG

- Um für alle Sprachräume einen einheitlichen Zeichencode zu haben, wurde Unicode entwickelt (Version 1.0 im Jahr 1991).
- Im Juni 2018 (Version 11.0.0) unterstützt Unicode 146 Schriften mit 137374 Codepoints, darunter 1212 Emojis.
- Organisiert in 17 Ebenen mit jeweils 2^{16} Codepoints (manche allerdings ungenutzt)

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Unicode

UNI FREIBURG

- Um für alle Sprachräume einen einheitlichen Zeichencode zu haben, wurde Unicode entwickelt (Version 1.0 im Jahr 1991).
- Im Juni 2018 (Version 11.0.0) unterstützt Unicode 146 Schriften mit 137374 Codepoints, darunter 1212 Emojis.
- Organisiert in 17 Ebenen mit jeweils 2^{16} Codepoints (manche allerdings ungenutzt)
- Die ersten 128 Codepoints stimmen mit ASCII überein, die ersten 256 mit ISO-Latin-1.

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Unicode

UNI FREIBURG

- Um für alle Sprachräume einen einheitlichen Zeichencode zu haben, wurde Unicode entwickelt (Version 1.0 im Jahr 1991).
- Im Juni 2018 (Version 11.0.0) unterstützt Unicode 146 Schriften mit 137374 Codepoints, darunter 1212 Emojis.
- Organisiert in 17 Ebenen mit jeweils 2^{16} Codepoints (manche allerdings ungenutzt)
- Die ersten 128 Codepoints stimmen mit ASCII überein, die ersten 256 mit ISO-Latin-1.
- Zum Thema Emojis gibt es ein eigenes Subkomitee ...

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

UTF-32, UTF-16 und UTF-8

UNI FREIBURG

- Ein Unicode-Zeichen kann durch eine 32-Bit-Zahl dargestellt werden (UTF-32 oder UCS-4).

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

UTF-32, UTF-16 und UTF-8

UNI FREIBURG

- Ein Unicode-Zeichen kann durch eine 32-Bit-Zahl dargestellt werden (UTF-32 oder UCS-4).
- Meist wird nur die Ebene 0 benötigt. Daher ist es effizienter, die Kodierung UTF-16 einzusetzen, bei der die Ebene 0 direkt als 16-Bit-Zahl kodiert wird. Zeichen aus anderen Ebenen benötigen 32 Bit.

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

UTF-32, UTF-16 und UTF-8

UNI FREIBURG

- Ein Unicode-Zeichen kann durch eine 32-Bit-Zahl dargestellt werden (UTF-32 oder UCS-4).
- Meist wird nur die Ebene 0 benötigt. Daher ist es effizienter, die Kodierung UTF-16 einzusetzen, bei der die Ebene 0 direkt als 16-Bit-Zahl kodiert wird. Zeichen aus anderen Ebenen benötigen 32 Bit.
- Im WWW wird meist UTF-8 eingesetzt:

Unicode	UTF-8 binär
0–127	0xxxxxxx
128–2047	110xxxxx 10xxxxxx
2048–65535	1110xxxx 10xxxxxx 10xxxxxx
65536–1114111	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

UTF-32, UTF-16 und UTF-8

UNI FREIBURG

- Ein Unicode-Zeichen kann durch eine 32-Bit-Zahl dargestellt werden (UTF-32 oder UCS-4).
- Meist wird nur die Ebene 0 benötigt. Daher ist es effizienter, die Kodierung UTF-16 einzusetzen, bei der die Ebene 0 direkt als 16-Bit-Zahl kodiert wird. Zeichen aus anderen Ebenen benötigen 32 Bit.
- Im WWW wird meist UTF-8 eingesetzt:

Unicode	UTF-8 binär
0–127	0xxxxxxx
128–2047	110xxxxx 10xxxxxx
2048–65535	1110xxxx 10xxxxxx 10xxxxxx
65536–1114111	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

- Wie kommen die komischen Zeichen auf Webseiten zustande?

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

UTF-32, UTF-16 und UTF-8

UNI FREIBURG

- Ein Unicode-Zeichen kann durch eine 32-Bit-Zahl dargestellt werden (UTF-32 oder UCS-4).
- Meist wird nur die Ebene 0 benötigt. Daher ist es effizienter, die Kodierung UTF-16 einzusetzen, bei der die Ebene 0 direkt als 16-Bit-Zahl kodiert wird. Zeichen aus anderen Ebenen benötigen 32 Bit.
- Im WWW wird meist UTF-8 eingesetzt:

Unicode	UTF-8 binär
0–127	0xxxxxxx
128–2047	110xxxxx 10xxxxxx
2048–65535	1110xxxx 10xxxxxx 10xxxxxx
65536–1114111	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

- Wie kommen die komischen Zeichen auf Webseiten zustande?
- Oft sind ISO-Latin-1/UTF-8 Verwechslungen der Grund!

Funktions-Aufrufe
Syntax
Standardfunktionen
Exkurs: Zeichenkodierung und Unicode
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

UNI FREIBURG

Funktions-Aufrufe

Mathematische Funktionen

math-Modul

Direktimport

Funktions-Definition

Namens-raum

Rückgabe-werte

Mathematische Funktionen

Mathematische Funktionen: Das Math-Modul

UNI FREIBURG

- Funktionen wie `sin` stehen nicht direkt zur Verfügung. Sie müssen durch Importieren des Mathematik-Moduls `math` bekannt gemacht werden.
- Werte aus dem Modul können durch Voranstellen von `math.` genutzt werden (Punktschreibweise):

Python-Interpreter

```
>>> import math
>>> math.pi
```


Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

Mathematische Funktionen: Das Math-Modul

UNI FREIBURG

- Funktionen wie `sin` stehen nicht direkt zur Verfügung. Sie müssen durch Importieren des Mathematik-Moduls `math` bekannt gemacht werden.
- Werte aus dem Modul können durch Voranstellen von `math.` genutzt werden (Punktschreibweise):

Python-Interpreter

```
>>> import math
>>> math.pi
3.141592653589793
>>>
```


Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

Mathematische Funktionen: Das Math-Modul

UNI FREIBURG

- Funktionen wie `sin` stehen nicht direkt zur Verfügung. Sie müssen durch Importieren des Mathematik-Moduls `math` bekannt gemacht werden.
- Werte aus dem Modul können durch Voranstellen von `math.` genutzt werden (Punktschreibweise):

Python-Interpreter

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.sin(1/4*math.pi)
```


Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

Mathematische Funktionen: Das Math-Modul

UNI FREIBURG

- Funktionen wie `sin` stehen nicht direkt zur Verfügung. Sie müssen durch Importieren des Mathematik-Moduls `math` bekannt gemacht werden.
- Werte aus dem Modul können durch Voranstellen von `math.` genutzt werden (Punktschreibweise):

Python-Interpreter

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.sin(1/4*math.pi)
0.7071067811865475
>>>
```


Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

Mathematische Funktionen: Das Math-Modul

UNI FREIBURG

- Funktionen wie `sin` stehen nicht direkt zur Verfügung. Sie müssen durch Importieren des Mathematik-Moduls `math` bekannt gemacht werden.
- Werte aus dem Modul können durch Voranstellen von `math.` genutzt werden (Punktschreibweise):

Python-Interpreter

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.sin(1/4*math.pi)
0.7071067811865475
>>> math.sin(math.pi)
```


Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

Mathematische Funktionen: Das Math-Modul

UNI FREIBURG

- Funktionen wie `sin` stehen nicht direkt zur Verfügung. Sie müssen durch Importieren des Mathematik-Moduls `math` bekannt gemacht werden.
- Werte aus dem Modul können durch Voranstellen von `math.` genutzt werden (Punktschreibweise):

Python-Interpreter

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.sin(1/4*math.pi)
0.7071067811865475
>>> math.sin(math.pi)
1.2246467991473532e-16
>>>
```


Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

Mathematische Funktionen: Das Math-Modul

UNI FREIBURG

- Funktionen wie `sin` stehen nicht direkt zur Verfügung. Sie müssen durch Importieren des Mathematik-Moduls `math` bekannt gemacht werden.
- Werte aus dem Modul können durch Voranstellen von `math.` genutzt werden (Punktschreibweise):

Python-Interpreter

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.sin(1/4*math.pi)
0.7071067811865475
>>> math.sin(math.pi)
1.2246467991473532e-16
>>> math.exp(math.log(2))
```


Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

Mathematische Funktionen: Das Math-Modul

UNI FREIBURG

- Funktionen wie `sin` stehen nicht direkt zur Verfügung. Sie müssen durch Importieren des Mathematik-Moduls `math` bekannt gemacht werden.
- Werte aus dem Modul können durch Voranstellen von `math.` genutzt werden (Punktschreibweise):

Python-Interpreter

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.sin(1/4*math.pi)
0.7071067811865475
>>> math.sin(math.pi)
1.2246467991473532e-16
>>> math.exp(math.log(2))
2.0
```


Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

Mathematische Funktionen: Direkt importieren

UNI FREIBURG

- Die Punktschreibweise verhindert Namenskollisionen, ist aber umständlich
- Direkter Import eines Bezeichners:
  `from` *module* `import` *name*
- Direkter Import aller Bezeichner eines Moduls:
  `from` *module* `import *`

Python-Interpreter

```
>>> from math import pi
>>> pi
```


Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

Mathematische Funktionen: Direkt importieren

UNI FREIBURG

- Die Punktschreibweise verhindert Namenskollisionen, ist aber umständlich
- Direkter Import eines Bezeichners:
  `from` *module* `import` *name*
- Direkter Import aller Bezeichner eines Moduls:
  `from` *module* `import *`

Python-Interpreter

```
>>> from math import pi
>>> pi
3.141592653589793
>>>
```

Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

Mathematische Funktionen: Direkt importieren

UNI FREIBURG

- Die Punktschreibweise verhindert Namenskollisionen, ist aber umständlich
- Direkter Import eines Bezeichners:
  `from` *module* `import` *name*
- Direkter Import aller Bezeichner eines Moduls:
  `from` *module* `import *`

Python-Interpreter

```
>>> from math import pi
>>> pi
3.141592653589793
>>> from math import *
>>> cos(pi)
```


Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

Mathematische Funktionen: Direkt importieren

UNI FREIBURG

- Die Punktschreibweise verhindert Namenskollisionen, ist aber umständlich
- Direkter Import eines Bezeichners:
  `from` *module* `import` *name*
- Direkter Import aller Bezeichner eines Moduls:
  `from` *module* `import *`

Python-Interpreter

```
>>> from math import pi
>>> pi
3.141592653589793
>>> from math import *
>>> cos(pi)
-1.0
```


Funktions-Aufrufe
Mathematische Funktionen
math-Modul
Direktimport
Funktions-Definition
Namensraum
Rückgabewerte

UNI FREIBURG

Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Funktionsdefinitionen

Neue Funktionen definieren

UNI FREIBURG

- Ein Python-Programm kann Funktionen selbst definieren.
- Eine Funktionsdefinition beginnt mit dem Schlüsselwort `def`, danach kommt der Funktionsname gefolgt von der Parameterliste und dann ein Doppelpunkt.
- Nach diesem Funktionskopf gibt der Python-Interpreter das Funktionsprompt-Zeichen `...` aus.
- Dann folgt der Funktionsrumpf: *Gleich weit eingerückte* Anweisungen, z.B. Zuweisungen oder Funktionsaufrufe:

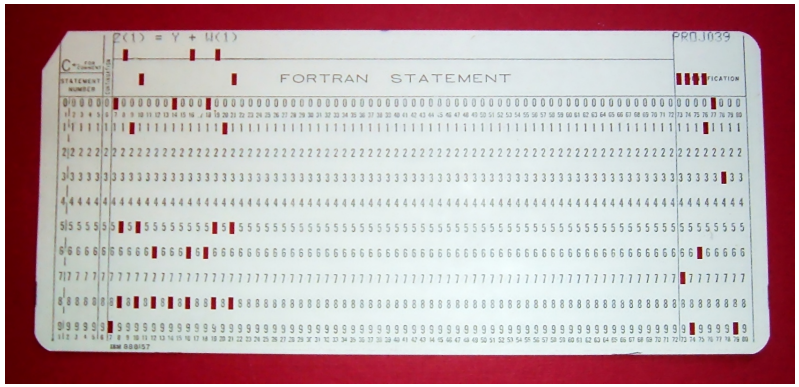
Python-Interpreter

```
>>> def print_lyrics():
...     print("I'm a lumberjack, and I'm okay")
...     print("I sleep all night and I work all day")
...
>>>
```


Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Definition
Einrückungen
Aufruf
Argumente, Parameter, Rückgabewerte
Namensraum
Rückgabewerte

Einrückungen in Python

- Einrückungen am Zeilenanfang sind bedeutungstragend.
  Vgl FORTRAN:



- Gleiche Einrückung = zusammengehöriger Block von Anweisungen
- In den meisten anderen Programmiersprachen durch Klammerung { } oder klammernde Schlüsselwörter.
- Wie viele Leerzeichen sollte man machen?

→ PEP8: 4 Leerzeichen pro Ebene (keine Tabs nutzen!)

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Definition
Einrückungen
Aufruf
Argumente, Parameter, Rückgabewerte
Namensraum
Rückgabewerte

Intermezzo: play the lumberjack

UNI FREIBURG

Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Python-Interpreter

```
>>> import webbrowser
>>>
webbrowser.open('https://www.youtube.com/watch?v=89LfQUlcNFk')
True
```

Selbst definierte Funktionen nutzen

UNI FREIBURG

- Funktionsnamen verhalten sich wie Variablennamen.
- Funktionen haben einen speziellen Typ.
- Selbstdefinierte Funktionen werden wie Standardfunktionen aufgerufen

Python-Interpreter

```
>>> print(print_lyrics)
```

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Definition
Einrückungen
Aufruf
Argumente, Parameter, Rückgabewerte
Namensraum
Rückgabewerte

Selbst definierte Funktionen nutzen

UNI FREIBURG

- Funktionsnamen verhalten sich wie Variablennamen.
- Funktionen haben einen speziellen Typ.
- Selbstdefinierte Funktionen werden wie Standardfunktionen aufgerufen

Python-Interpreter

```
>>> print(print_lyrics)
<function print_lyrics at 0x100520560>
```


Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Selbst definierte Funktionen nutzen

UNI FREIBURG

- Funktionsnamen verhalten sich wie Variablennamen.
- Funktionen haben einen speziellen Typ.
- Selbstdefinierte Funktionen werden wie Standardfunktionen aufgerufen

Python-Interpreter

```
>>> print(print_lyrics)
<function print_lyrics at 0x100520560>
>>>
```


Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Selbst definierte Funktionen nutzen

UNI FREIBURG

- Funktionsnamen verhalten sich wie Variablennamen.
- Funktionen haben einen speziellen Typ.
- Selbstdefinierte Funktionen werden wie Standardfunktionen aufgerufen

Python-Interpreter

```
>>> print(print_lyrics)
<function print_lyrics at 0x100520560>
>>> type(print_lyrics)
```


Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Selbst definierte Funktionen nutzen

UNI FREIBURG

- Funktionsnamen verhalten sich wie Variablennamen.
- Funktionen haben einen speziellen Typ.
- Selbstdefinierte Funktionen werden wie Standardfunktionen aufgerufen

Python-Interpreter

```
>>> print(print_lyrics)
<function print_lyrics at 0x100520560>
>>> type(print_lyrics)
<class 'function'>
```


Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Selbst definierte Funktionen nutzen

UNI FREIBURG

- Funktionsnamen verhalten sich wie Variablennamen.
- Funktionen haben einen speziellen Typ.
- Selbstdefinierte Funktionen werden wie Standardfunktionen aufgerufen

Python-Interpreter

```
>>> print(print_lyrics)
<function print_lyrics at 0x100520560>
>>> type(print_lyrics)
<class 'function'>
>>>
```


Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Selbst definierte Funktionen nutzen

UNI FREIBURG

- Funktionsnamen verhalten sich wie Variablennamen.
- Funktionen haben einen speziellen Typ.
- Selbstdefinierte Funktionen werden wie Standardfunktionen aufgerufen

Python-Interpreter

```
>>> print(print_lyrics)
<function print_lyrics at 0x100520560>
>>> type(print_lyrics)
<class 'function'>
>>> print_lyrics()
```


Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Selbst definierte Funktionen nutzen

UNI FREIBURG

- Funktionsnamen verhalten sich wie Variablennamen.
- Funktionen haben einen speziellen Typ.
- Selbstdefinierte Funktionen werden wie Standardfunktionen aufgerufen

Python-Interpreter

```
>>> print(print_lyrics)
<function print_lyrics at 0x100520560>
>>> type(print_lyrics)
<class 'function'>
>>> print_lyrics()
I'm a lumberjack, and I'm okay
I sleep all night and I work all day
```


Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Selbst definierte Funktionen nutzen

UNI FREIBURG

- Funktionsnamen verhalten sich wie Variablennamen.
- Funktionen haben einen speziellen Typ.
- Selbstdefinierte Funktionen werden wie Standardfunktionen aufgerufen

Python-Interpreter

```
>>> print(print_lyrics)
<function print_lyrics at 0x100520560>
>>> type(print_lyrics)
<class 'function'>
>>> print_lyrics()
I'm a lumberjack, and I'm okay
I sleep all night and I work all day
>>>
```


Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Selbst definierte Funktionen nutzen

UNI FREIBURG

- Funktionsnamen verhalten sich wie Variablennamen.
- Funktionen haben einen speziellen Typ.
- Selbstdefinierte Funktionen werden wie Standardfunktionen aufgerufen

Python-Interpreter

```
>>> print(print_lyrics)
<function print_lyrics at 0x100520560>
>>> type(print_lyrics)
<class 'function'>
>>> print_lyrics()
I'm a lumberjack, and I'm okay
I sleep all night and I work all day
>>> print_lyrics = 42
```


Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Definition
Einrückungen
Aufruf
Argumente, Parameter, Rückgabewerte
Namensraum
Rückgabewerte

Definierte Funktionen in Funktionsdefinitionen

UNI FREIBURG

Was passiert hier?

Python-Interpreter

```
>>> def print_lyrics():
...     print("I'm a lumberjack, and I'm okay")
...     print("I sleep all night and I work all day")
...
>>>
```


Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Definierte Funktionen in Funktionsdefinitionen

UNI FREIBURG

Was passiert hier?

Python-Interpreter

```
>>> def print_lyrics():
...     print("I'm a lumberjack, and I'm okay")
...     print("I sleep all night and I work all day")
...
>>>
>>> def repeat_lyrics():
...     print_lyrics()
...     print_lyrics()
...
>>>
```


Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Definierte Funktionen in Funktionsdefinitionen

UNI FREIBURG

Was passiert hier?

Python-Interpreter

```
>>> def print_lyrics():
...     print("I'm a lumberjack, and I'm okay")
...     print("I sleep all night and I work all day")
...
>>>
>>> def repeat_lyrics():
...     print_lyrics()
...     print_lyrics()
...
>>> repeat_lyrics()
I'm a lumberjack ...
```

Was wird hier exakt ausgeführt?

Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Definition

Einrückungen

Aufruf

Argumente, Parameter, Rückgabewerte

Namensraum

Rückgabewerte

Argumente, Parameter und Rückgabewerte

UNI FREIBURG

- Selbst definierte Funktionen benötigen oft *Argumente*.
- Die Definition verwendet *formale Parameter* (Variablennamen), an die beim Aufruf die *Argumentwerte* zugewiesen werden.
- `return` beendet die Ausführung der Funktion.
- Der Wert des Ausdrucks nach `return` wird zum Wert des Funktionsaufrufs.

Python-Interpreter

```
>>> CM_PER_INCH = 2.54
>>> def cm_to_inches(l_cm):
...     return l_cm / CM_PER_INCH
...
>>> cm_to_inches(195)
76.77165354330708
```


Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Definition
Einrückungen
Aufruf
Argumente, Parameter, Rückgabewerte
Namensraum
Rückgabewerte

UNI FREIBURG

Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Namensraum

Lokale Variablen und Parameter

Kellertabelle

Traceback

Globale Variablen

Rückgabewerte

Namensraum

Namensraum von lokalen Variablen und Parametern

UNI FREIBURG

- Parameter sind nur innerhalb der Funktion sichtbar.
- Lokal (durch Zuweisung) eingeführte Variablen ebenfalls.

Python-Interpreter

```
>>> def cat_twice(part1, part2):
...     cat = part1 + part2
...     print(cat)
...     print(cat)
...
>>>
```


Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Lokale Variablen und Parameter
Kellertabelle
Traceback
Globale Variablen
Rückgabewerte

Namensraum von lokalen Variablen und Parametern

UNI FREIBURG

- Parameter sind nur innerhalb der Funktion sichtbar.
- Lokal (durch Zuweisung) eingeführte Variablen ebenfalls.

Python-Interpreter

```
>>> def cat_twice(part1, part2):
...     cat = part1 + part2
...     print(cat)
...     print(cat)
...
>>> line1 = 'Bing tiddle '
>>> line2 = 'tiddle bang.'
>>> cat_twice(line1, line2)
```


Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Lokale Variablen und Parameter
Kellertabelle
Traceback
Globale Variablen
Rückgabewerte

Namensraum von lokalen Variablen und Parametern

UNI FREIBURG

- Parameter sind nur innerhalb der Funktion sichtbar.
- Lokal (durch Zuweisung) eingeführte Variablen ebenfalls.

Python-Interpreter

```
>>> def cat_twice(part1, part2):
...     cat = part1 + part2
...     print(cat)
...     print(cat)
...
>>> line1 = 'Bing tiddle '
>>> line2 = 'tiddle bang.'
>>> cat_twice(line1, line2)
Bing tiddle tiddle bang.
Bing tiddle tiddle bang.
>>>
```


Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Lokale Variablen und Parameter
Kellertabelle
Traceback
Globale Variablen
Rückgabewerte

Namensraum von lokalen Variablen und Parametern

UNI FREIBURG

- Parameter sind nur innerhalb der Funktion sichtbar.
- Lokal (durch Zuweisung) eingeführte Variablen ebenfalls.

Python-Interpreter

```
>>> def cat_twice(part1, part2):
...     cat = part1 + part2
...     print(cat)
...     print(cat)
...
>>> line1 = 'Bing tiddle '
>>> line2 = 'tiddle bang.'
>>> cat_twice(line1, line2)
Bing tiddle tiddle bang.
Bing tiddle tiddle bang.
>>> cat
```


Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Lokale Variablen und Parameter
Kellertabelle
Traceback
Globale Variablen
Rückgabewerte

Namensraum von lokalen Variablen und Parametern

UNI FREIBURG

- Parameter sind nur innerhalb der Funktion sichtbar.
- Lokal (durch Zuweisung) eingeführte Variablen ebenfalls.

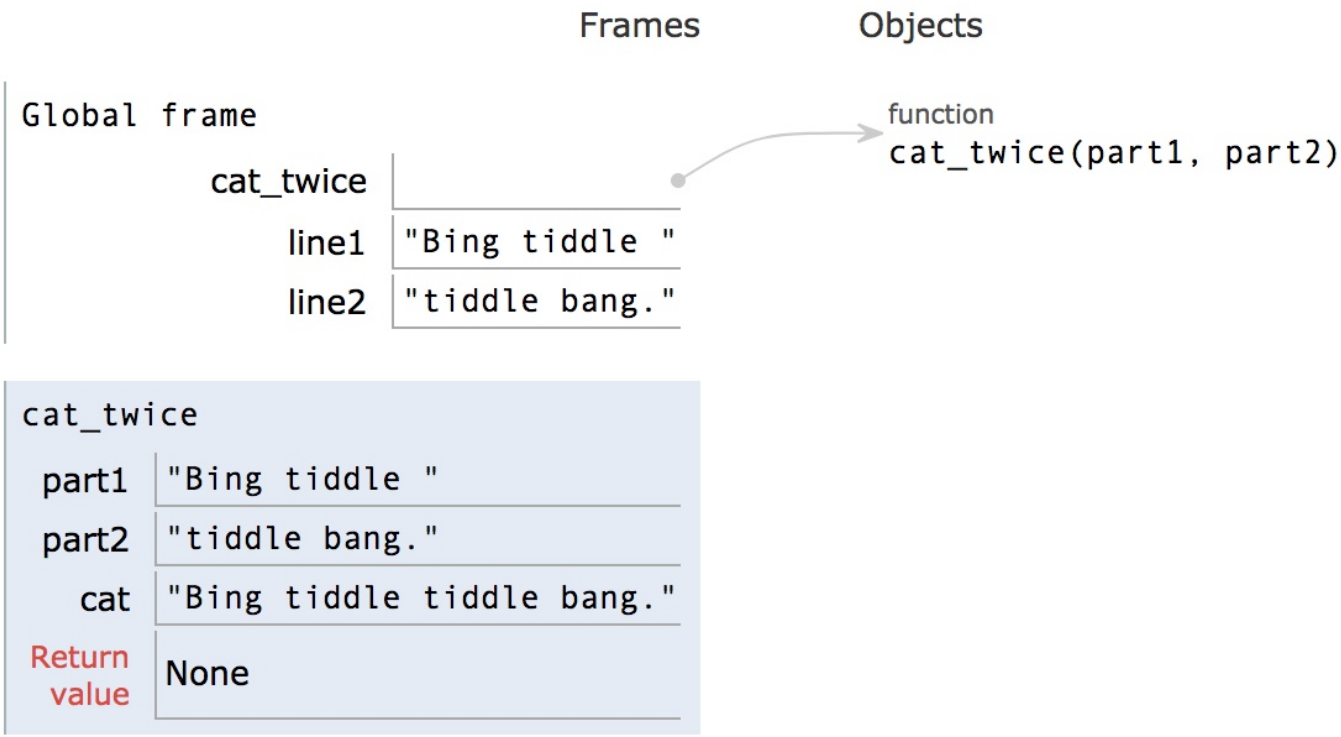
Python-Interpreter

```
>>> def cat_twice(part1, part2):
...     cat = part1 + part2
...     print(cat)
...     print(cat)
...
>>> line1 = 'Bing tiddle '
>>> line2 = 'tiddle bang.'
>>> cat_twice(line1, line2)
Bing tiddle tiddle bang.
Bing tiddle tiddle bang.
>>> cat
NameError: name 'cat' is not defined
```


Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Lokale Variablen und Parameter
Kellertabelle
Traceback
Globale Variablen
Rückgabewerte

Kellertabelle

- Die Variablenbelegungen innerhalb von Funktionsaufrufen können durch eine Kellertabelle visualisiert werden (hier hilft `pythontutor.com`).
  Ende von `cat_twice`



Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Lokale Variablen und Parameter
Kellertabelle
Traceback
Globale Variablen
Rückgabewerte

Traceback

UNI FREIBURG

- Tritt bei der Ausführung einer Funktion ein Fehler auf, z.B. Zugriff auf die nicht vorhandene Funktion `print_twice` in `cat_twice`, dann gibt es ein Traceback (entsprechend einer Kellertabelle):

Python-Interpreter

```
>>> def cat_twice(part1, part2):
...     cat = part1 + part2
...     print_twice(cat)
...
>>> cat_twice('foo ', 'bar!')
```


Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Lokale Variablen und Parameter
Kellertabelle
Traceback
Globale Variablen
Rückgabewerte

Traceback

UNI FREIBURG

- Tritt bei der Ausführung einer Funktion ein Fehler auf, z.B. Zugriff auf die nicht vorhandene Funktion `print_twice` in `cat_twice`, dann gibt es ein Traceback (entsprechend einer Kellertabelle):

Python-Interpreter

```
>>> def cat_twice(part1, part2):
...     cat = part1 + part2
...     print_twice(cat)
...
>>> cat_twice('foo ', 'bar!')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 3, in cat_twice
NameError: name 'print_twice' is not defined
```


Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Lokale Variablen und Parameter
Kellertabelle
Traceback
Globale Variablen
Rückgabewerte

Globale Variablen

UNI FREIBURG

- Funktionen sollen vorrangig lokale Variable und Parameter nutzen.
- Funktionen können globale Variablen lesen, falls es keine lokale Variable gleichen Namens gibt.

Python-Interpreter

```
>>> master = 666
>>> def slave():
...     return master
>>> print("slave returns", slave())
>>> def independent(master):
...     return master
>>> print("independent returns", independent(42))
>>> def ignorant():
...     master = 333
...     return master
>>> print("ignorant returns", ignorant(), "master=", master)
```


Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Lokale Variablen und Parameter
Kellertabelle
Traceback
Globale Variablen
Rückgabewerte

UNI FREIBURG

Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Namensraum

Rückgabewerte

Rückgabewerte

Rückgabewerte

UNI FREIBURG

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

- Alle Funktionen geben einen Wert zurück.
- Funktionen wie `print` geben einen speziellen Wert `None` zurück, der nicht angezeigt wird.

Python-Interpreter

```
>>> result = print('Bruce')
Bruce
>>> result
>>> print(result)
None  [das ist nicht der String 'None'!]
```

- `None` ist der einzige Wert des Typs `NoneType`.

Einen Wert zurück geben

UNI FREIBURG

- Das Schlüsselwort `return` erlaubt die Definition des Rückgabewerts.

Python-Interpreter

```
>>> def sum3(a, b, c):
...     return a + b + c
...
>>> sum3(1, 2, 3)
6
```

- Funktionen ohne `return` (wie `cat_twice`) geben `None` zurück.

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

return ≠ print

UNI FREIBURG

- Können wir nicht auch `print(·)` benutzen, um einen Funktionswert zurück zu geben?

Python-Interpreter

```
>>> def printsum3(a, b, c):
...     print(a + b + c)
...
>>> sum3(1, 2, 3)
6
```


Funktions-Aufrufe

Mathematische Funktionen

Funktions-Definition

Namensraum

Rückgabewerte

return ≠ print

UNI FREIBURG

- Können wir nicht auch `print(·)` benutzen, um einen Funktionswert zurück zu geben?

Python-Interpreter

```
>>> def printsum3(a, b, c):
...     print(a + b + c)
...
>>> sum3(1, 2, 3)
6
>>> sum3(1, 2, 3) + 4
```

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

return ≠ print

UNI FREIBURG

- Können wir nicht auch `print(·)` benutzen, um einen Funktionswert zurück zu geben?

Python-Interpreter

```
>>> def printsum3(a, b, c):
...     print(a + b + c)
...
>>> sum3(1, 2, 3)
6
>>> sum3(1, 2, 3) + 4
10
```

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

return ≠ print

UNI FREIBURG

- Können wir nicht auch `print(·)` benutzen, um einen Funktionswert zurück zu geben?

Python-Interpreter

```
>>> def printsum3(a, b, c):
...     print(a + b + c)
...
>>> sum3(1, 2, 3)
6
>>> sum3(1, 2, 3) + 4
10
>>> printsum3(1, 2, 3) + 4
```

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

return ≠ print

UNI FREIBURG

- Können wir nicht auch `print(·)` benutzen, um einen Funktionswert zurück zu geben?

Python-Interpreter

```
>>> def printsum3(a, b, c):
...     print(a + b + c)
...
>>> sum3(1, 2, 3)
6
>>> sum3(1, 2, 3) + 4
10
>>> printsum3(1, 2, 3) + 4
6
TypeError: unsupported operand type(s) for +:
'NoneType' and 'int'
```

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Zusammenfassung

UNI FREIBURG

- Funktionen sind benannte vorgegebene Programmstücke (Standardfunktionen) oder selbst definierte Funktionen.

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Zusammenfassung

UNI FREIBURG

- Funktionen sind benannte vorgegebene Programmstücke (Standardfunktionen) oder selbst definierte Funktionen.
- Beim Aufruf einer Funktion müssen Argumente angegeben werden, die die formalen Parameter mit Werten belegen.

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Zusammenfassung

UNI FREIBURG

- Funktionen sind benannte vorgegebene Programmstücke (Standardfunktionen) oder selbst definierte Funktionen.
- Beim Aufruf einer Funktion müssen Argumente angegeben werden, die die formalen Parameter mit Werten belegen.
- Funktionen geben normalerweise einen Funktionswert zurück, der mit `return` festgelegt wird.

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Zusammenfassung

UNI FREIBURG

- Funktionen sind benannte vorgegebene Programmstücke (Standardfunktionen) oder selbst definierte Funktionen.
- Beim Aufruf einer Funktion müssen Argumente angegeben werden, die die formalen Parameter mit Werten belegen.
- Funktionen geben normalerweise einen Funktionswert zurück, der mit `return` festgelegt wird.
- Funktionen führen einen neuen Namensraum für die Parameter und lokalen Variablen (durch Zuweisung eingeführt) ein.

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Zusammenfassung

UNI FREIBURG

- Funktionen sind benannte vorgegebene Programmstücke (Standardfunktionen) oder selbst definierte Funktionen.
- Beim Aufruf einer Funktion müssen Argumente angegeben werden, die die formalen Parameter mit Werten belegen.
- Funktionen geben normalerweise einen Funktionswert zurück, der mit `return` festgelegt wird.
- Funktionen führen einen neuen Namensraum für die Parameter und lokalen Variablen (durch Zuweisung eingeführt) ein.
- Globale Variablen können gelesen werden, falls sie nicht durch einen Parameter oder eine lokale Variable überdeckt werden.

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte

Zusammenfassung

UNI FREIBURG

- Funktionen sind benannte vorgegebene Programmstücke (Standardfunktionen) oder selbst definierte Funktionen.
- Beim Aufruf einer Funktion müssen Argumente angegeben werden, die die formalen Parameter mit Werten belegen.
- Funktionen geben normalerweise einen Funktionswert zurück, der mit `return` festgelegt wird.
- Funktionen führen einen neuen Namensraum für die Parameter und lokalen Variablen (durch Zuweisung eingeführt) ein.
- Globale Variablen können gelesen werden, falls sie nicht durch einen Parameter oder eine lokale Variable überdeckt werden.
- `pythontutor.com` visualisiert die Programmausführung mit Hilfe von Kellertabellen.

Funktions-Aufrufe
Mathematische Funktionen
Funktions-Definition
Namensraum
Rückgabewerte