

A cartoon green snake with a yellow underbelly and a friendly expression is positioned on the left side of the slide. It has a large white speech bubble with a black outline coming from its mouth. The background is a photograph of a London street scene featuring Big Ben, the London Eye, and a red double-decker bus. A semi-transparent red banner is at the bottom.


Programmieren
lernen mit Python

Funktionen mit Parametern

Kira Grammel, Nina Ihde, Sebastian Serth & Selina Reinhard
Hasso-Plattner-Institut
Universität Potsdam

Funktionsdefinition mit Parametern

```
1 def funktions_name(aufgelistete Parameter):  
2     # Anweisungen
```



Einer Funktion können mehrere Parameter übergeben werden.
Diese werden mit einem Komma voneinander getrennt.

Parameter und Argumente

- Parameter sind Variablen in der Funktionsdefinition
- Beim Funktionsaufruf werden Argumente mitgegeben, dies sind tatsächliche Werte für die Parameter
- Beispiel, das wir oft genutzt haben: `print("Hallo")`
- Statt "Hallo" haben wir auch viele andere Argumente mitgegeben

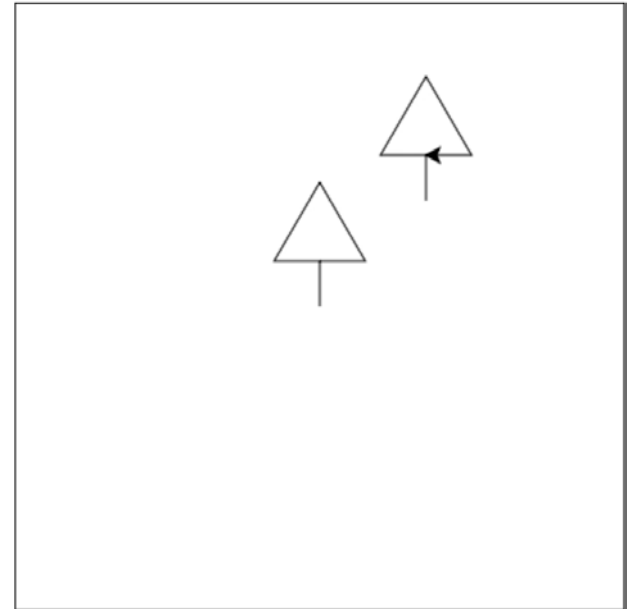
Baum an den Koordinaten (x, y) zeichnen

```

1 from turtle import *
2 def baum(x,y):
3     up()
4     goto(x,y)
5     down()
6     setheading(90)
7     forward(30)
8     left(90)
9     forward(30)
10    right(120)
11    forward(60)
12    right(120)
13    forward(60)
14    right(120)
15    forward(30)
16
17    baum(0,0)
18    baum(70,70)

```

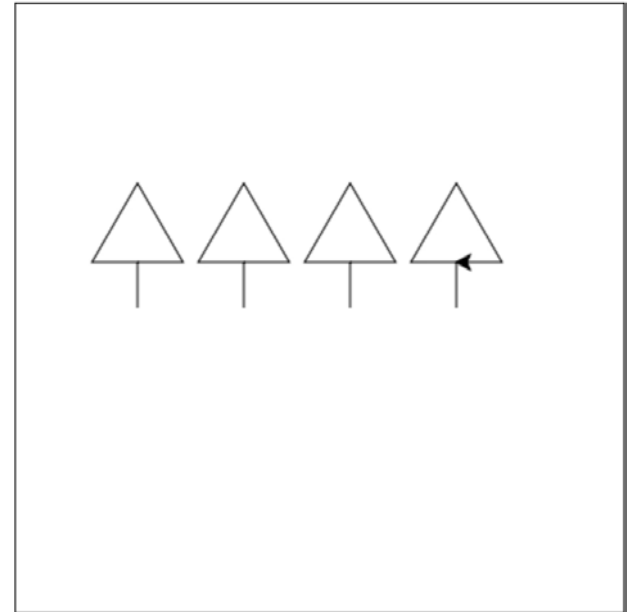
Verwendung von den mitgegebenen Parametern x und y



Zwei Funktionsaufrufe mit verschiedenen Argumenten

Funktionsaufruf in for-Schleife

```
1 from turtle import *
2 for i in [-120, -50, 20, 90]:
3     baum(i, 0)
```



- Die Funktion `baum()` wird vier Mal aufgerufen mit den Werten aus der Liste
- Der Funktionsaufruf muss die gleiche Anzahl an Argumenten enthalten und die Reihenfolge ist entscheidend

- Schreibe eine Funktion, die aus einer Liste mit Attraktionen in London, die ermittelt, die mit einem B anfangen.
- Liefere das Ergebnis wieder als Liste von Attraktionen.



Anfangsbuchstaben überprüfen

```
1 def startet_mit_b(attraktion):  
2     ergebnis = attraktion[0] == "B"  
3     return ergebnis
```

Teilaufgabe 1

- Ermittle den Anfangsbuchstaben einer Attraktion
- `attraktion[0]` gibt den Anfangsbuchstaben zurück
- `attraktion[0] == "B"` überprüft, ob der Anfangsbuchstabe gleich B ist und gibt `True` oder `False` zurück

Anfangsbuchstaben überprüfen

```
1 def startet_mit_b(attraktion):  
2     ergebnis = attraktion[0] == "B"  
3     return ergebnis  
4  
5 big_ben_mit_b = startet_mit_b("Big Ben")  
6 print(big_ben_mit_b)  
7 print(startet_mit_b("London Eye"))
```

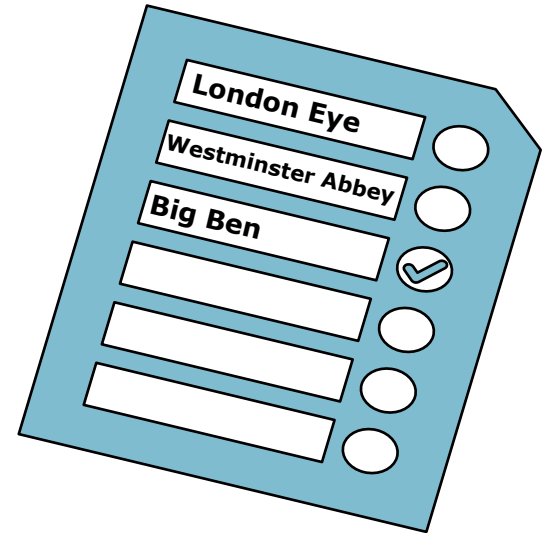
True
False

Teilaufgabe 1

- Funktionsaufruf mit Argument "Big Ben" ergibt somit True
- Und Funktionsaufruf mit "London Eye" ergibt False

Liste mit ausgewählten Attraktionen erstellen

```
1 def attraktionen_mit_b(liste):  
2     ergebnis = []  
3     for attraktion in liste:  
4         if startet_mit_b(attraktion):  
5             ergebnis.append(attraktion)  
6     return ergebnis
```



Teilaufgabe 2

- Baue eine Liste aus Attraktionen auf, die mit "B" anfangen
- Leere Liste: `ergebnis = []`
- Element in Liste aufnehmen mit Funktion `append`
 - `ergebnis.append(neuer_Wert)`

Alle Attraktionen mit "B"

```
1 attraktionen = ["London Eye", "Big Ben", "Palace of Westminster", "Westminster Abbey", "St.  
Margaret's Church", "Buckingham Palace", "Victoria Memorial", "St. James's Park", "Trafalgar Square", "National  
Gallery", "St. Pauls Cathedral", "British Library", "Kings Cross Station", "Notting Hill"]  
2  
3 def startet_mit_b(attraktion):  
4     ergebnis = attraktion[0] == "B"  
5     return ergebnis  
6  
7 def attraktionen_mit_b(liste):  
8     ergebnis = []  
9     for attraktion in liste:  
10         if startet_mit_b(attraktion):  
11             ergebnis.append(attraktion)  
12     return ergebnis  
13  
14 print(attraktionen_mit_b(attraktionen))
```

```
['Big Ben', 'Buckingham Palace', 'British Library']
```

Zwei Parameter

```

1 attraktionen = ["London Eye", "Big Ben", "Palace of Westminster", "Westminster Abbey", "St.
Margaret's Church", "Buckingham Palace", "Victoria Memorial", "St. James's Park", "Trafalgar Square", "National
Gallery", "St. Pauls Cathedral", "British Library", "Kings Cross Station", "Notting Hill"]
2
3 def startet_mit(attraktion, buchstabe):
4     ergebnis = attraktion[0] == buchstabe
5     return ergebnis
6
7 def attraktionen_mit(liste, buchstabe):
8     ergebnis = []
9     for attraktion in liste:
10         if startet_mit(attraktion, buchstabe):
11             ergebnis.append(attraktion)
12     return ergebnis
13
14 print(attraktionen_mit(attraktionen, "L"))

```

```
['London Eye']
```

Zwei Parameter

```
1 attraktionen = ["London Eye", "Big Ben", "Palace of Westminster", "Westminster Abbey", "St.  
Margaret's Church", "Buckingham Palace", "Victoria Memorial", "St. James's Park", "Trafalgar Square", "National  
Gallery", "St. Pauls Cathedral", "British Library", "Kings Cross Station", "Notting Hill"]  
2  
3 def startet_mit(attraktion, buchstabe):  
4     ergebnis = attraktion[0] == buchstabe  
5     return ergebnis  
6  
7 def attraktionen_mit(liste, buchstabe):  
8     ergebnis = []  
9     for attraktion in liste:  
10         if startet_mit(attraktion, buchstabe):  
11             ergebnis.append(attraktion)  
12     return ergebnis  
13  
14 print(attraktionen_mit(attraktionen, "N"))
```

```
['National Gallery', 'Notting Hill']
```

Zusammenfassung

- Bei der Funktionsdefinition können in Klammern mit Kommata getrennt Parameter mitgegeben werden
- Diese werden beim Funktionsaufruf durch tatsächliche Werte - Argumente - ersetzt
- `def funktions_name(aufgelistete Parameter):`
 # Anweisungen

