Control structures & Memory

Jannusch Bigge

07.11.2023

Memory

Memory

Different types of memory:

HDD/SSD Hard Disk Drive / Solid State Drive

Memory

Different types of memory:

HDD/SSD	Hard Disk Drive / Solid State Drive
RAM	Random Access Memory

Memory

Different types of memory:

HDD/SSD	Hard Disk Drive / Solid State Drive
RAM	Random Access Memory
Cache	Small and fast memory

Memory - Reference and copy

Python

- Reference: $b = a$

presudo memory

Memory - Reference and copy

Python

- Reference: $b = a$

presudo memory

a (00)	00 03
b (01)	00 (address of a)

Memory - Reference and copy

Python

- Reference: $b = a$
- Copy: $a = b.copy()$

presudo memory

a (00) 00 03

b (01) 00 (address of a)

Memory - Reference and copy

Python

- Reference: $b = a$
- Copy: $a = b.copy()$

presudo memory

a (00) 00 03

b (01) 00 (address of a)

–

a (00) 00 03

b (01) 00 03

Scopes and Namespaces

Using every name only ones could lead to problems.

Scopes and Namespaces

Using every name only ones could lead to problems.

Solution:

Scopes and Namespaces

Using every name only ones could lead to problems.

Solution:

- Namespaces

Scopes and Namespaces

Using every name only ones could lead to problems.

Solution:

- Namespaces
- Scopes

Namespaces

Mapping from names to objects:

Namespaces

Mapping from names to objects:

- Built-in

Namespaces

Mapping from names to objects:

- Built-in
- Global

Namespaces

Mapping from names to objects:

- Built-in
- Global
- Enclosing
- Local

Scope - Single definition

```
>>> x = 'global'
>>> def foo():
...
...     def bar():
...         print(x)
...
...     bar()
>>> foo()
global
```

Scope - Double definition

```
>>> x = 'global'
>>> def foo():
...     x = 'enclosing'
...     def bar():
...         print(x)
...
...     bar()
>>> foo()
enclosing
```

Scope - Triple definition

```
>>> x = 'global'
>>> def foo():
...     x = 'enclosing'
...     def bar():
...         x = 'local'
...         print(x)
...
...     bar()
>>> foo()
local
```

Control structures

Control structures

Only sequential execution is nice but sometimes we need more:

6th fibonacci number:

```
first = 0
second = 1
third = 1
fourth = 2
fifth = 3
sixth = 5
```

Control structures

Only sequential execution is nice but sometimes we need more:

6th fibonacci number:

```
first = 0
second = 1
third = 1
fourth = 2
fifth = 3
sixth = 5
```

Problem: What if we want the 100th fibonacci number?

Control structures

Only sequential execution is nice but sometimes we need more:

6th fibonacci number:

```
first = 0
second = 1
third = 1
fourth = 2
fifth = 3
sixth = 5
```

Problem: What if we want the 100th fibonacci number?

Solution: Control structures

```
a = 0
b = 1
c = 1
for i in range(100):
    a = b
    b = c
    c = a + b
```

for loop

Repeat something a given number of times.

- **for** *variable* **in** *iterable*:
- **for** *variable* **in** *range(start, stop, step)*:

for loop

Repeat something a given number of times.

- **for** *variable* **in** *iterable*:
- **for** *variable* **in** *range(start, stop, step)*:

Example:

```
      a = 2
      for i in range(3):
          print(a + i)
>>> 2
>>> 3
>>> 4
```

if statement

Do something only if condition is true.

- **if** *condition*:

if statement

Do something only if condition is true.

- **if** *condition*:

Types of conditions:

- **True** or **False**
- **==** or **!=**
- **and** or **or**

if statement

Do something only if condition is true.

- **if** *condition*:

Types of conditions:

- **True** or **False**
- **==** or **!=**
- **and** or **or**

Example:

```
    a = 2
    if a == 2:
        print(a)
>>> 2
```

```
a = 2
if a == 2 and a != 3:
    print('Not␣3')
>>> 'Not␣3'
```

Task

First Task:

- Just try the fibonacci number sequence for different numbers.

Second Task:

1. Calculate a baseline
   1.1 Calculate the mean of each cluster
   1.2 Calculate the mean of all points
   1.3 Sort the unclassified points into the two clusters
2. Clustering with Gram-Schmidt (Bonus Task)
   2.1 Find the perpendicular vector to the line between (-2, 6) and (6, -2)
   2.2 Find the threshold **t** for $w^T x < t$ when x is in class one. Note: $w^T x = w \cdot x$

Next week: Using functions and libraries