Python deel 2

Types die we al wisten

- Boolean (waar/niet waar = True)
- Integer (getal = 10)
- Float (getal = 10.05)
- String (woord=“tekst“)

list

- List is een array van getallen of tekst.
- Deze kan aangepast worden in de code.

- leden = ["Sim Soony", "Marry Roundknee", "Jack Corridor"]
- jaartallen = [1970, 1967, 1960, 1850]

- `print(leden[1])`
- `jaartallen[0] = 1971`

dictionary

- Dictionary zijn paren van **keys** en **values**:

```
phone_numbers = {
      "John Smith": "+37682929928",
      "Marry Simpons": "+423998200919
}
```

dictionary

- Dictionary zijn paren van **keys** en **values**:

```
tekst_naar_nummer = {
      "een": 1,
      "twee": 2,
      "drie": 3
}
print(tekst_naar_nummer["een"])
tekst_naar_nummer["vier"] = 4
```

Pas het raadspel van vorige week aan: Vraag niet om een cijfer als input, maar vraag om tekst (bijvoorbeeld “twee”) en zet die met een dictionary om naar een getal.

Wiskunde

commando	actie	voorbeeld
+	optellen	1 + 1 = 2
-	aftrekken	4 - 3 = 1
*	vermenigvuldigen	2 * 2 = 4
/	delen	6 / 2 = 3

Wiskunde

commando	actie	voorbeeld
+	optellen	1 + 1 = 2
-	aftrekken	4 - 3 = 1
*	vermenigvuldigen	2 * 2 = 4
/	delen	6 / 2 = 3
//	vloerdeling (naar beneden afronden)	7 // 2 = 3 (2***3** + 1)

Wiskunde

commando	actie	voorbeeld
+	optellen	1 + 1 = 2
-	aftrekken	4 - 3 = 1
*	vermenigvuldigen	2 * 2 = 4
/	delen	6 / 2 = 3
//	vloerdeling (naar beneden afronden)	7 // 2 = 3 (2*3 + 1)
%	modulo (rest)	7 % 2 = 1 (2*3 + 1)

Wiskunde

commando	actie	voorbeeld
+	optellen	1 + 1 = 2
-	aftrekken	4 - 3 = 1
*	vermenigvuldigen	2 * 2 = 4
/	delen	6 / 2 = 3
//	vloerdeling (naar beneden afronden)	7 // 2 = 3 (2*__3__ + 1)
%	modulo (rest)	7 % 2 = 1 (2*3 + __1__)
**	exponent	2 ** 3 = (2^3) = 8

If elif else

```
nummer = 3

if nummer % 2 == 0:
    print("even")
else:
    print("oneven")
```

operator	function
<	kleiner dan
<=	kleiner dan of gelijk aan
>	groter dan
>=	groter dan of gelijk aan
==	gelijk aan
!=	verschillend van
<>	een alternatief voor verschillend van

Maak een nieuw programma dat om een nummer vraagt en print of het “even” of “oneven” is.

If elif else

```
leeftijd = int(input("Wat is je leeftijd?"))
if leeftijd < 18:
    print("Je bent nog niet volwassen!")
elif leeftijd < 30:
    print("Je bent nog jong!")
elif leeftijd < 50:
    print("Beginnen grijze haren te komen?")
else:
    print("Wegen de jaren zwaar?")
```

Loops

```
while True
```

```
nummer = 0
while nummer < 5:
    print(nummer)
    nummer = nummer + 1
```

```
for nummer in [0,1,2,3,4]:
    print(nummer)
```

```
for nummer in range(0,5):
    print(nummer)
```

For loop

```
lijst = [3,6,5,613,7,8,9]

for cijfer in lijst:
    if cijfer % 2 == 0:
        print("even")
    else:
        print("oneven")
```

For loop met continue

```
lijst = [3,6,5,613,7,8,9]

for cijfer in lijst:
    if cijfer % 2 == 0:
        print("even")
    else:
        print("oneven")
```

```
for cijfer in lijst:
    if cijfer % 2 == 0:
        print("even")
        continue

    print("oneven")
```

ProgrammeerClub Challenge

- Schrijf een programma dat alle getallen van 1 tot 100 print.
- Bij elke veelvoud van **3** moet het programma '**Programmeer**' printen in plaats van het getal
- Bij elke veelvoud van **5** moet je '**Club**' printen.
- Een veelvoud van **3 en 5** print je als '**ProgrammeerClub**‘.
- Probeer het programma zoveel mogelijk in één keer op te schrijven voordat je het test.

FizzBuzz

- https://en.wikipedia.org/wiki/Fizz_buzz

Libraries

- In de terminal: `pip install telwoord`

```
D:\workspaces\various\vughts-programmeerclub>pip install telwoord
Collecting telwoord
  Downloading telwoord-0.4.tar.gz (3.8 kB)
Using legacy 'setup.py install' for telwoord, since package 'wheel' is not installed.
Installing collected packages: telwoord
    Running setup.py install for telwoord ... done
Successfully installed telwoord-0.4
```

Libraries

- In de terminal: `pip install telwoord`

```
from telwoord import cardinal
```

Libraries

- In de terminal: `pip install telwoord`

```
from telwoord import cardinal

nummer = int(input("Geef een nummer op: "))
```

Libraries

- In de terminal: `pip install telwoord`

```
from telwoord import cardinal

nummer = int(input("Geef een nummer op: "))

print(cardinal(nummer))
```

Libraries

- In de terminal: `pip install telwoord`

```
from telwoord import cardinal

nummer = int(input("Geef een nummer op: "))

print(cardinal(nummer))
```

Maak dit programma na

Libs met als voorbeeld Pygame

- pip install pygame
- pip3 install pygame

- Gebruiken met import …. als eerste in je code (import pygame)

Programma‘s delen

- https://pyinstaller.readthedocs.io/en/stable/

Python 3 Beginner's Reference Cheat Sheet

Alvaro Sebastian
http://www.sixthresearcher.com

Main data types

boolean = *True* / *False*
integer = 10
float = 10.01
string = "123abc"
list = [value1, value2, ...]
dictionary = { key1:value1, key2:value2, ...}

Numeric operators

+	addition
-	subtraction
*	multiplication
/	division
**	exponent
%	modulus
//	floor division

Comparison operators

==	equal
!=	different
>	higher
<	lower
>=	higher or equal
<=	lower or equal

Boolean operators

and	logical AND
or	logical OR
not	logical NOT

Special characters

#	coment
\n	new line
\<char>	scape char

String operations

string[i]	retrieves character at position i
string[-1]	retrieves last character
string[i:j]	retrieves characters in range i to j

List operations

list = []	defines an empty list
list[i] = x	stores x with index i
list[i]	retrieves the item with index I
list[-1]	retrieves last item
list[i:j]	retrieves items in the range i to j
del list[i]	removes the item with index i

Dictionary operations

dict = {}	defines an empty dictionary
dict[k] = x	stores x associated to key k
dict[k]	retrieves the item with key k
del dict[k]	removes the item with key k

String methods

string.upper()	converts to uppercase
string.lower()	converts to lowercase
string.count(x)	counts how many times x appears
string.find(x)	position of the x first occurrence
string.replace(x,y)	replaces x for y
string.strip(x)	returns a list of values delimited by x
string.join(L)	returns a string with L values joined by string
string.format(x)	returns a string that includes formatted x

List methods

list.append(x)	adds x to the end of the list
list.extend(L)	appends L to the end of the list
list.insert(i,x)	inserts x at i position
list.remove(x)	removes the first list item whose value is x
list.pop(i)	removes the item at position i and returns its value
list.clear()	removes all items from the list
list.index(x)	returns a list of values delimited by x
list.count(x)	returns a string with list values joined by S
list.sort()	sorts list items
list.reverse()	reverses list elements
list.copy()	returns a copy of the list

Dictionary methods

dict.keys()	returns a list of keys
dict.values()	returns a list of values
dict.items()	returns a list of pairs (key,value)
dict.get(k)	returns the value associtated to the key k
dict.pop()	removes the item associated to the key and returns its value
dict.update(D)	adds keys-values (D) to dictionary
dict.clear()	removes all keys-values from the dictionary
dict.copy()	returns a copy of the dictionary

Legend: x,y stand for any kind of data values, **s** for a string, **n** for a number, **L** for a list where **i,j** are list indexes, **D** stands for a dictionary and **k** is a dictionary key.