B: variables

Programming with Python

- Simple and *minimal-chic* programming language
- Easy to learn and use: reading code is almost like reading English
- Simplicity as a result of design:
  - focus on the problem and not on the language/the data representation in memory
  - a way to represent an easy-to-read program
- Python is free, open source well-supported and well-maintaned:
  - VS Code, Jupyter, PIP, PyPi, Anaconda, Docker...
- Comes from "Monty Python"

Programming in Python

- Sequence of one or more instructions
- A fixed alphabet, strict syntactical rules and notations
- high-level computer language
- The Python interpreter maps Py. commands, one after the other, into low-level commands
- Computer hardware understands and executes the low-level commands
  - Even adding two numbers requires a certain visibility of RAM and CPU

```
load the number from memory location 2001 into the
CPU

load the number from memory location 2002 into the
CPU

add the two numbers in the CPU

Store the result into location 2003
```

Programming

- Python is easy to understand
- interpreters translate high-level Py. to machine language
- Two ways to do it, using:
    - Compiler or
    - Interpreter

```
grand_total = price + vat

print(“Welcome to Python”)
```

Simple Python commands

- Comments!
  - Use of hashtag **#**
- Print command **print**
  - Prints on screen!
    - Numbers and operations

```
# Command to print number 10!
print(10)

# More commands...
print(10+20)
print(10+2*20)
print((2*4)+6)
print(2**2)
```

Simple Python commands

- **print()**
  - Prints on screen!
  - Text is *containerized* by means of single (‘’) or double (“”) quotes

```
# prints on the screen:

print(‘Have a nice Autumn term!’)
```

Simple Python commands

- **print()**
  - Prints on screen!
  - Text is *containerized* by means of single (‘’) or double (“”) quotes

```
# Let’s print two names
print(“Stelios”)
print(“Sotiriadis”)
# Or
print(“Stelios”, “Sotiriadis”)

# Or
print(‘Stelios’, ‘Sotiriadis’)
# Or
print(‘Stelios ’+‘Sotiriadis’)
```

Variables

- Programs manipulate data that sits in the computer memory.
- Variables are generic names for the container of some value
- Computer variables are akin to both parameters and variables of mathematics
- A funny ‘=’ symbol to assign value to a variable

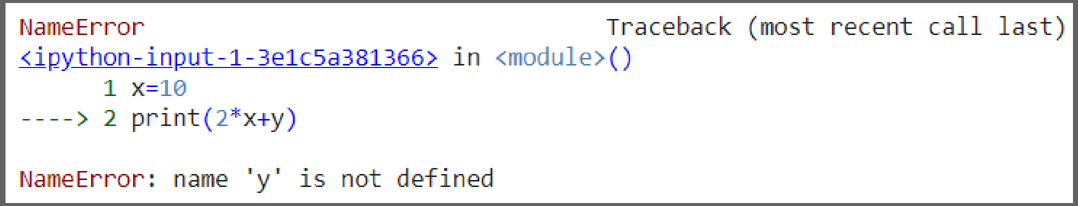
```
name = `Nik`
age = 10
```

```
print(name)
```

```
x = age + 10
print(x)
```

```
print(10/x) # beware division by 0!
```

```
print(2*x + y) # What does it print?
```



```
NameError                                 Traceback (most recent call last)
<ipython-input-1-3e1c5a381366> in <module>()
      1 x=10
----> 2 print(2*x+y)

NameError: name 'y' is not defined
```

Variables can change their content

- Assignment statement
    - `fahrenheit = 9/5 * celsius + 32`

```
# This is an assignment expression!
myvar = 10
print(myvar)

myvar = 20 # Assign a new value
print(myvar)

myvar = myvar+1
print(myvar) # Prints 21
```

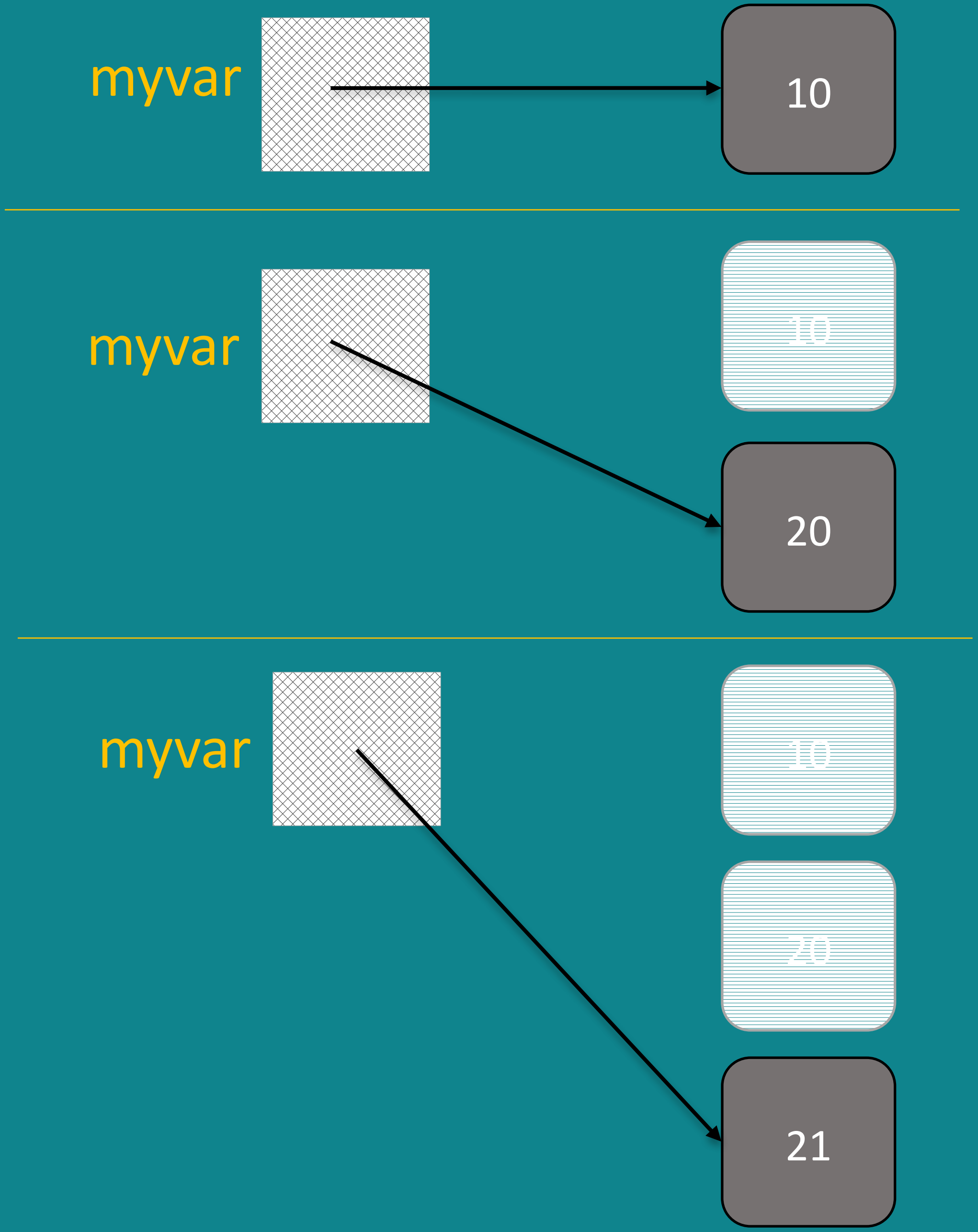
Variables can change!

```
# This is an assignment expression!
myvar = 10
print(myvar)

myvar = 20 # Assign a new value
print(myvar)

myvar = myvar+1
print(myvar) # Prints 21
```

How memory looks like!



Programming is about input/output

- Input from the keyboard:
    - ‘enter’/’return’ to end the input.

      input()

    - Assign the input to a variable

      <variable> = input(<prompt>)

```
# This is an input example
name = input("Please enter a name: ")
print(name)


# This is another input example
my_fav_num = input(“What is your favourite number? ”)
print(“Your favourite number is ”, my_fav_num)
```

Programming is about input/output

- Input from the keyboard:
  - ‘enter’/’return’ to end the input.

    input()

  - Assign the input to a variable

    <variable> = input(<prompt>)

```
# This is another input example
fName = input(“Give first name: ”)
lName = input(“Give surname: ”)
print(fName, lName)
```

Quiz 2: Fill the gaps

a) ______________________ # Write a statement to print your name

b) ______________________ # Write a statement to print number 10

c) ______________________ # Write a statement to print the sum of 1 and 2

Quiz 2 Solution

a) **print(“Stelios”)** # Write a statement to print your name

b) **print(10)** # Write a statement to print number 10

c) **print(1+2)** Write a statement to print the sum of 1 and 2

Quiz 3: Fill the gaps

```
# Provide the command to prompt the user to enter a name

my_input = d) ________________(“Enter your name: “)


# Provide the command to print the variable my_input

e) ________________ (my_input)
```

Quiz 3 Solution

```
# Provide the command to prompt user to enter a name

my_input = d) __ input__(“Enter your name: “)


# Provide the command to print the variable my_input

e) __print__(my_input)
```