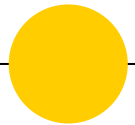
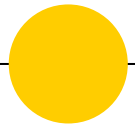
Python Fundamentals & Syntax



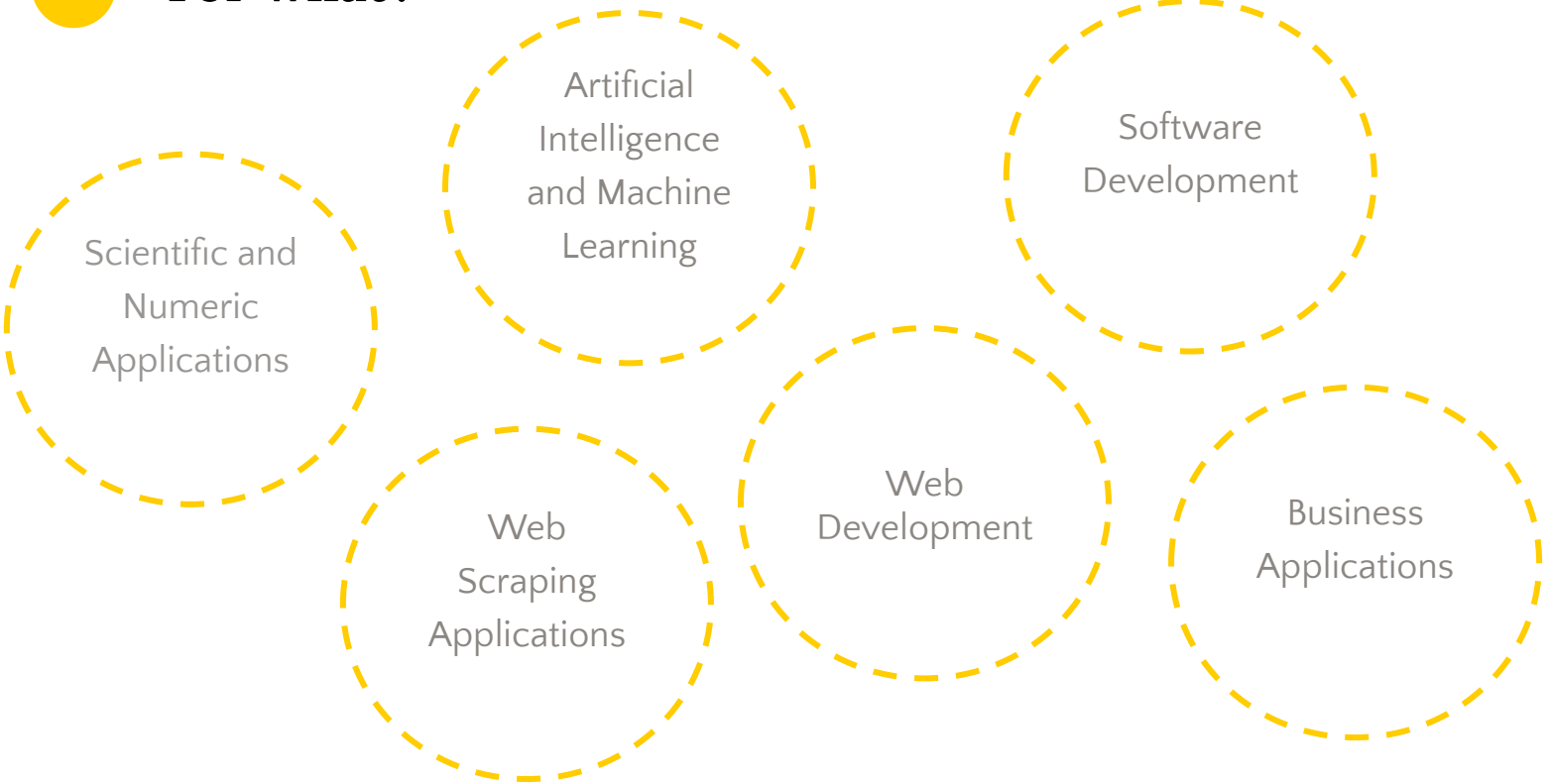
Introduction to Python



Why Python?

- Easy to learn
- (For us): data science library
- Object oriented, functional, interpreted
- Less lines of code to perform the same task as compared to other major languages like C/C++ and Java
- Dynamically typed
- Open sourced-> lots of resources

For what?


Artificial Intelligence and Machine Learning
Software Development
Scientific and Numeric Applications
Web Development
Business Applications
Web Scraping Applications

Who’s using Python?
(the cool kids or everyone?)

- Instagram is coded in Python
- it is one of the main languages used by engineers at Google
- Netflix uses Python to develop its recommendation algorithms
- Dropbox's desktop application is developed in Python
- Reddit is coded in Python

How to Install Python?

https://www.anaconda.com/products/distribution

https://code.visualstudio.com/download

Variables

Back to python and code !!!

Where should we write our code?

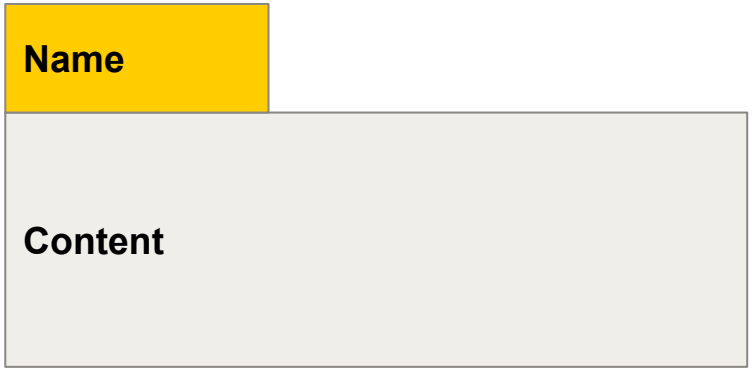
Writing python code

Jupyter Notebook | Terminal | VSCO/Spyder

Executing the same code, different ways

```
print(‘Hello”)
print(2+3))
```

Variables what is it



- Type
- Size
- address

Variables what is it

Types of Variables

Object Type	Example
Numbers	1234, Decimal(), Fraction()
Strings	‘hello’
Lists	[1, 2] [1, 4, ‘hello’] [1, 2, [1,2]]
Boolean	True False
Dictionaries	{“brand”:adidas , “amount”:3}
None	None
Tuple	(1, 2, ‘hello’)

Types of Variables

Object Type	Example
Sets	set(1,2,3)
Files	Text.txt, hello.py
Program unit types	Functions, module, class
Implementation-related types	Compiled code

Ah-hoc Types?

Introducing the basic types

Boolean

- True, False and 0 and 1
- Logical operators (OR, AND, + and * , not ..)

Numbers

- Type of numbers
  - Int, float ..
- Operations (+, -, >, <)

Numbers

- **Integer**
  - Normal and long (depends if python 2 or 3)
- **Float**
  - 1.2, 4,56
- **Others bases**
  - Hexadecimal etc
- **Complex numbers**
  - 3 + 4j

Numbers

- **Integer**
  - Normal and long (depends if python 2 or 3)
- **Float**
  - 1.2, 4,56
- **Others bases**
  - Hexadecimal etc
- **Complex numbers**
  - 3 + 4j

Strings

- Use for text information
- Sequence (positional ordering)

- Special string characters (\n)

Saving the value ?

```
a = ‘hello’
type(a)
```

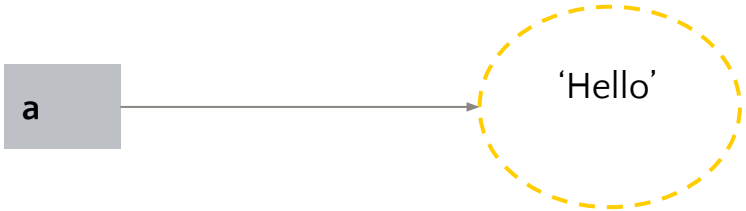
- a is the name of the variable
- The type is a string
- The content is hello

Variables

- Variables are created when they are first assigned values
- Variables are replaced with their values when used in expressions.
- Variables must be assigned before they can be used in expressions
- Variables refer to objects and are never declared ahead of time.

Variables

- Create an object to represent the value “Hello’
- Create the variable a, if it does not yet exist.
- Link the variable a to the new string object



Saving the value ?

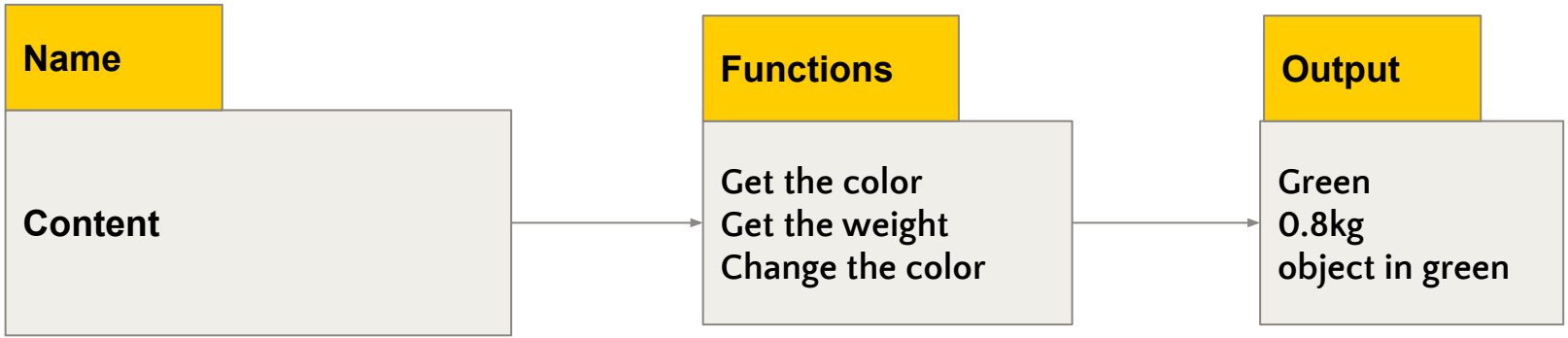
```
a, b = ‘hello’ , 4
print(a)

print(b)
```

Methods and built-in functions

- Operations all their own, *methods*—functions are available for a given object because of the object's type
  - A method for a string might not work for a number

Variables what is it



Saving the output?

Methods and built-in functions

```
len('hello') -> 5
```

```
len(a) -> 5
```

```
len('a') -> 1
```

```
print(a)
```

Others Methods

On string

```
upper, lower, capitalize, count, endswith, find,
format, isalnum, islower, isnumeric…replace, split,
strip, startswith, title, rstrip
```

Others Methods

On numbers

```
min, max, pow, abs, round, int, hex
```

F string

Creating customisable strings with our variables

Operators

Computing some operations

Built-in tools and Operators

Expression operator

+, -, *, /, >>, **, &, etc.

Built-in mathematical functions

pow, abs, round, int, hex, bin, etc.

Expression

Expression

- Combination of numbers and operators

A + B → result is another object

Numerical Operators

- +
- *
- **
- (-)
- %
- //
- /

Convert euros to pound
Olympic Pool in Liters: (50,25,2)

More Operators in the math module

- Import our first module !!
  - Use sqrt, cos, sin, e, inf, pi, nan
  - Floor, trunc, ceil

Derivate

- The module of (a + bj).
- sin(π/6)
- cos(2π)
- Check for some numbers that
    - eˆ(i * b) = cos(b) + i * sin(b).

Some operations .... Let's do them together

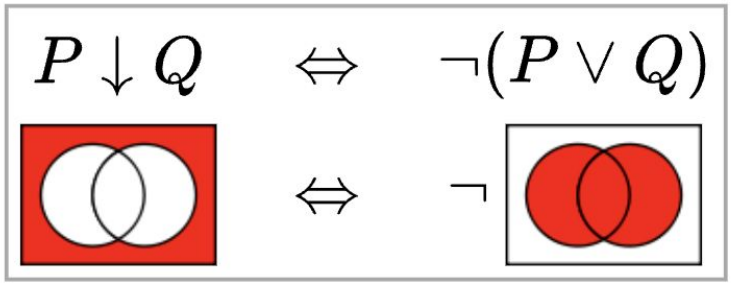
Increment operators (and Decrement)

- Increment a variable by 1 with +=
- Decrement with -=
- Possible with * and / too

Logic Operators

- OR
- AND
- NOT
- IN (for arrays)
- NOT IN (for arrays)
- IS
- IS NOT

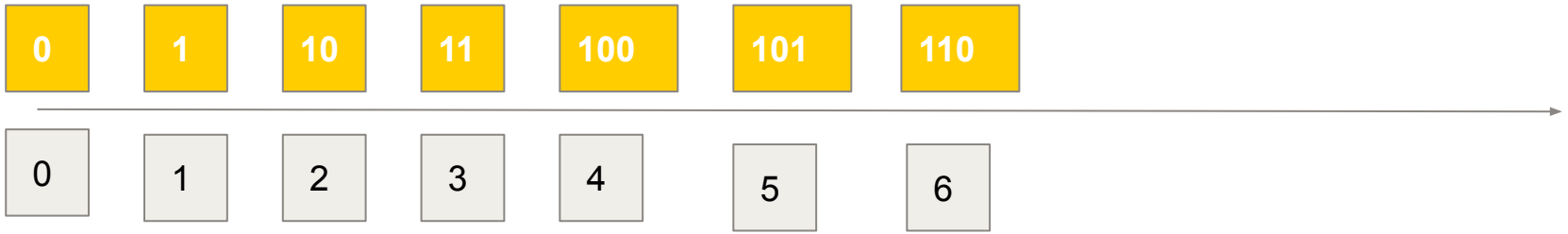
Build the logical NOR *operator*



Logical again!

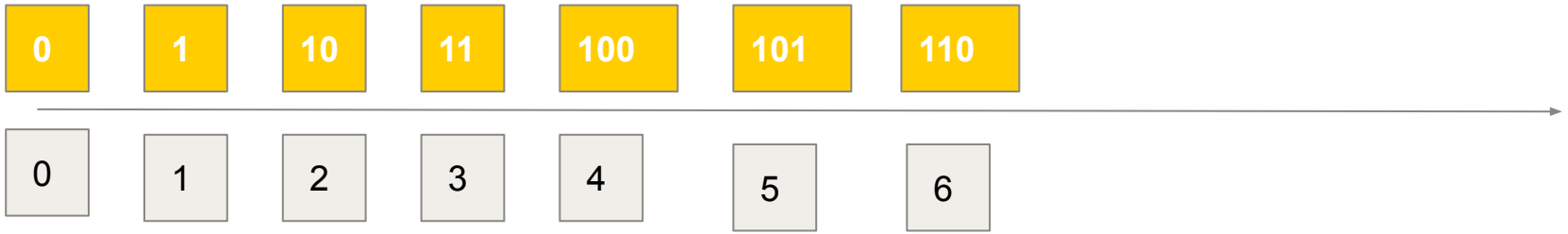
- A bit on the binary base
  - Bin() in python

Logical again!



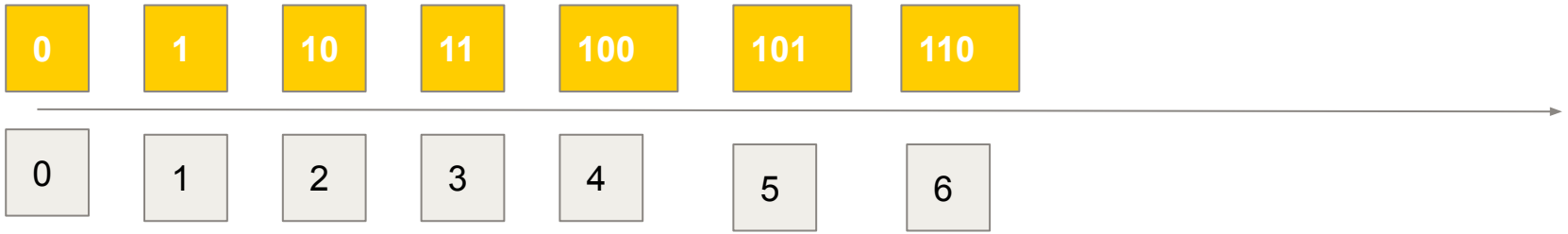
- With python
  - convert 0b101 in integer base 10
  - Convert 10 in base 2

Logical again!



- With python
  - convert 0b101 in integer base 10
  - Convert 10 in base 2

Logical again!



- Bitwise operator
  - << Bitwise left shift
  - >> Bitwise right shift

Comparison Relational Operators

- <, >, <=
- ==
- != (<>possible in python 2)
- | for sets
- & for sets

Operators, functions

- Lambda to define some operations to be performed
  - Practical in different cases
  - Anonymous function

  More details on it later ... :)

Operators Precedence?

- Operations can be chained
- How does python know which operation to perform first?
  - Parentheses if ‘no priority’
  - Type of the answer is the less complex object

Small Note on formatting the code

Conditional statements

If you feel like it, else not

If statement

- Compound statement
  - Combination of statements

- Program
  - If and its friends
  - But also Python syntax

What is a if statement

General form

```
if test1:
    do   # our first associated block
elif test2:                 # elif is optional
    do
else:
    Do # third statement
```

If statement

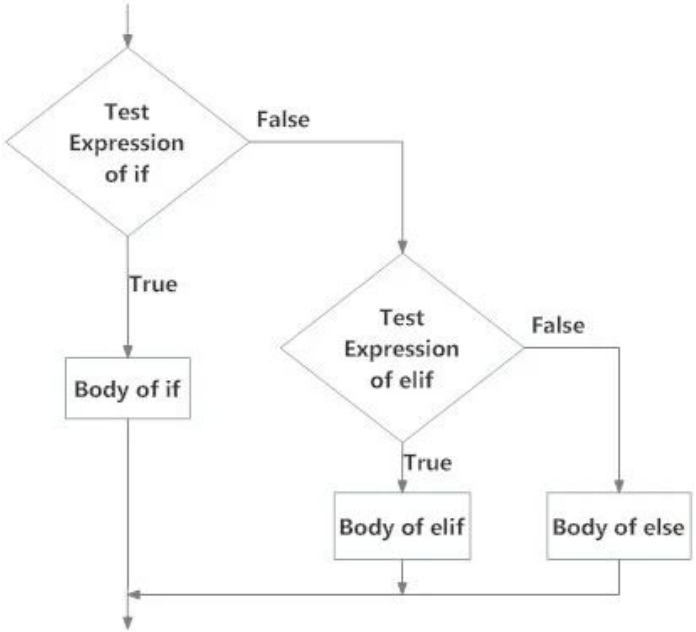


Fig: Operation of if...elif...else statement

Handling error

```
try:
    Do sth
except:
    Do sth else
```

Blocks and things

- Execute is linear, with if you can jump blocks
- Blocks and statement boundaries are detected automatically. No need of brackets
- Compound statements = header + “:” + indented statements.
- Blank lines, spaces, and comments are usually ignored.
- Docstrings are ignored but are saved and displayed by tools.

Indentation rule

```
x = 1
if x:
    y = 2
    if y == 3:
        print('block2')
    print('block1')

print('block0')
```

```
x = 1
if x:
    y = 2
    if y == 3:
    print('block2')
    print('block1')

print('block0')
```

Statement Delimiters

Lines and continuation

- (), {}, []
- Backlash \
- “” ””
- #

As a value?

A = Y if X else Z