Python Tutorial

Prepared By
Eng./Ahmed Abd Elrahman

What is Python

Python is a versatile and beginner-friendly programming language that is widely used for web development, data analysis, artificial intelligence, and more. This tutorial will guide you through the basics of

It is used for:

- web development (server-side),software development, Data science
- Python can be used on a server to create web applications.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

Python

Installing Python

Before you start coding, you need to have Python installed on your computer.- *Windows*:

Download the installer from the

[official Python website](https://www.python.org/downloads/), and follow the installation instructions.

Variable and Data Types

```
# Variables
x = 10
y = "Hello, World!“
z = 3.14

# Data Types
print(type(x))  # <class 'int'>
print(type(y))  # <class 'str'>
print(type(z))  # <class 'float'>
```

Comments

```
#### Comments in python
# This is a single-line comment
"""This is
amulti-line
Comment
"""
```

Casting

```
x = int(1)   # x will be 1
y = int(2.8) # y will be 2
z = int("3") # z will be 3


x = float(1)     # x will be 1.0
y = float(2.8)   # y will be 2.8
z = float("3")   # z will be 3.0


x = str("s1") # x will be 's1'
y = str(2)    # y will be '2'
```

Strings

- Strings in python are surrounded by either single quotation marks, or double quotation marks. 'hello' is the same as "hello".

Strings are Arrays

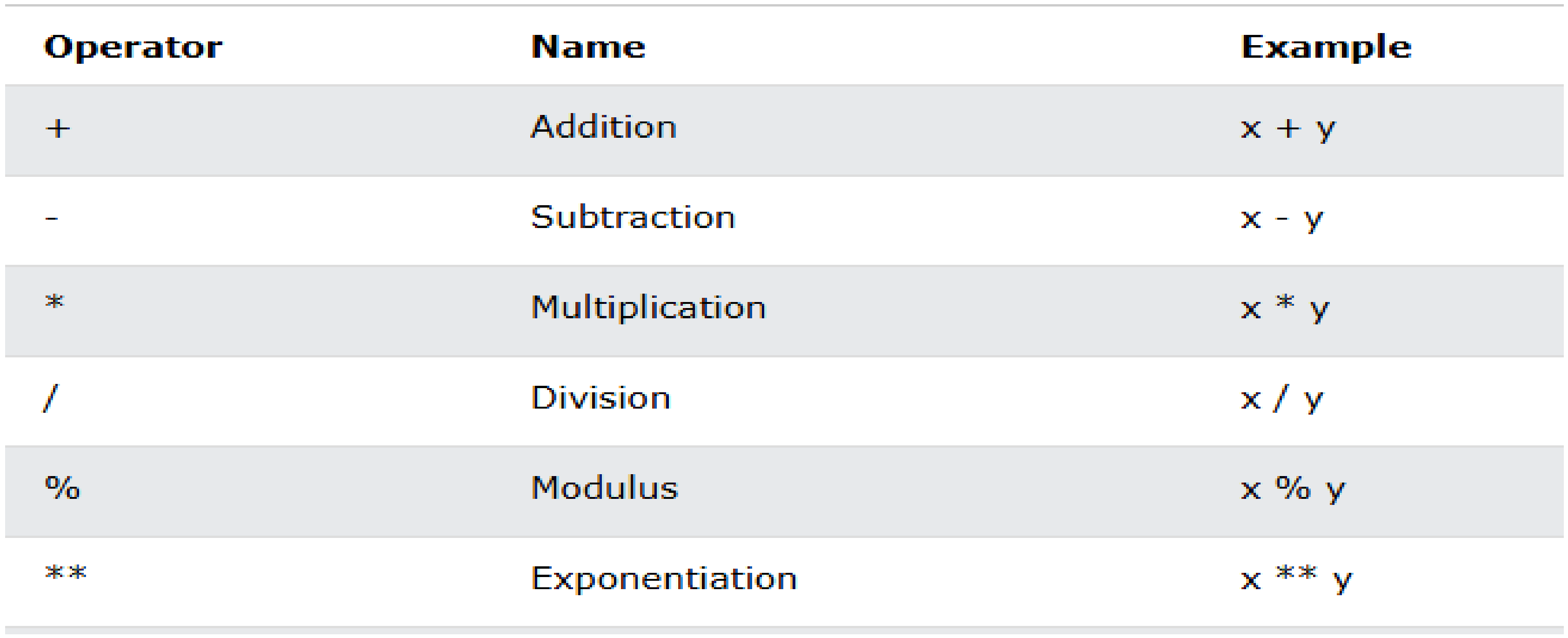
```
a = "Hello, World!"
print(a[1])
```

- To get the length of a string, use the **len()** function.
- To check if a certain phrase or character is present in a string, we can use the keyword **in**.
- txt = "The best things in life are Water!"
  print(“Water" **in** txt)   #True
-

- Booleans represent one of two values:
  True or **False**.

print(10 > 9)
print(10 == 9)
print(10 < 9)

-

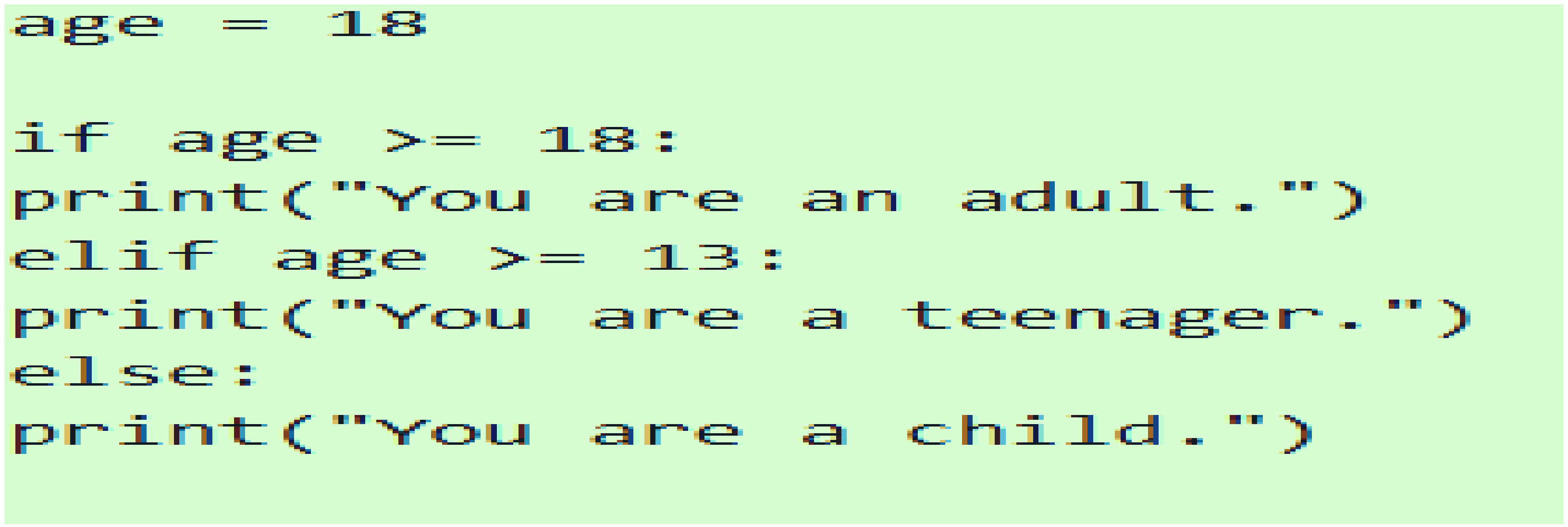


Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3
:=	print(x := 3)	x = 3 print(x)

Control Structures

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
else:
  print(“a is greater than b")
```



```
age = 18

if age >= 18:
print("You are an adult.")
elif age >= 13:
print("You are a teenager.")
else:
print("You are a child.")
```

Loops

#while

```
i = 1
while i < 6:
  print(i)
  i += 1
```

.break , With the break statement we can stop the loop even if the while condition is true:

```
i = 1
while i < 6:
  print(i)
  if i == 3:
    break
    i += 1
```

Loops

- **#continue,** With the continue statement we can stop the current iteration, and continue with the next:

```
i = 0
while i < 6:
  i += 1
  if i == 3:
    continue
  print(i)
```

Loops

```
  print(x)
0
1
2
3
4
5
for x in range(6):
  if x == 3: break
  print(x)
```

Functions

- In Python a function is defined using the **def** keyword:

```
def my_function():
  print("Hello from a function")
my_function()


def my_function(fname):
  print(fname + " Refsnes")

my_function("Emil")
my_function("Tobias")
my_function("Linus")
```

Functions

```
def add(x):
  return 10 + x

print(add(3))
print(add(5))
print(add(9))
```

Lists

- Lists are used to store multiple items in a single variable.

```
List1= [“asd", “ali", “omar"]
print(List1)
print(List1[0]) # ’asd’
print(len(thislist)) #3
```

- List items can be of any data type:

```
list1 = ["apple", "banana", "cherry"]
list2 = [1, 5, 7, 9, 3]
list3 = [True, False, False]
```

Lists

- A list can contain different data types:-

```
list1 = ["abc", 34, True, 40, "male"]
```

- To add an item to the end of the list, use the **append**() method

```
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)
```

- The **remove**() method removes the specified item.

```
List2= ["apple", "banana", "cherry"]
List2.remove("banana")
print(List2)
```

Lists

- **Clear** the list content:

```
List2= ["apple", "banana", "cherry"]
List2.clear()
print(List2)
```

#Loop Through a List

```
List3= ["apple", "banana", "cherry"]
for x in  List3 :
  print(x)


List3 = ["apple", "banana", "cherry"]
for i in range(len(List3)):
  print(List3[i])
```

Dictionaries

- Dictionaries are used to store data values in key: value pairs.

```
Dict2= {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(Dict2[“year"])  # 1964
```

- print(len(Dict2))

Dictionaries

```
# Adding a new key-value pair
Dict2["email"] = ‘alice@example.com’



# Looping through a dictionary
for key, value in Dict2.items():
  print(key+”,”+ value)
```

Array

- An array is a special variable, which can hold more than one value at a time.

```
cars = ["Ford", "Volvo", "BMW"]
print(len(cars))  # 3
print(cars[0])  # “Ford”
cars[0]=“Volly”
print(cars[0])  # “Volly”
```

# Looping Array Elements

```
for x in cars:
  print(x)
```

Error Handling

- Use try-except blocks to handle errors gracefully.
- The **try** block lets you test a block of code for errors.
- The **except** block lets you handle the error.
- The **finally** block lets you execute code, regardless of the result of the try- and except blocks.

Error Handling

- **The try block will generate an exception, because x is not defined:**

```
try:
  print(x)
except:
  print("An exception occurred")
```

- The **finally** block, if specified, will be executed regardless if the try block raises an error or not.

```
try:
  print(x)
except:
  print("Something went wrong")
finally:
  print("The 'try except' is finished")
```

Modules and Packages

- Python has a rich ecosystem of libraries. You can import and use them in your code.

```
import math
print(math.sqrt(16))  # 4.0

# Importing specific functions
from math import pi
print(pi)  # 3.141592653589793
```