

Hello Python!

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. ([https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)))

Where can you work with Python?

- `$ python3`
- `script.py + $ python3 script.py`
- in Jupyter Lab using notebooks

Documentation and resources:

- <https://docs.python.org/3/library/index.html>
- <https://devdocs.io/>
- <https://pythonfordesigners.com/>
- `SHIFT + TAB` (to read Python documentation inline)
- `TAB` (to get a list of options inline)

variables

```
In [126]: a = 3
          b = 2
          a * b
Out[127]: 6
```

```
In [128]: b + b
```

```
Out[128]: 4
```

You can store the result of an arithmetic in another variable, for example:

```
In [129]: c = a + b
```

```
In [130]: c
```

```
Out[130]: 5
```

print()

You can see the result of your arithmetic expression by directly running a cell, or, you can use the built-in print function.

```
In [131]: a
```

```
Out[131]: 3
```

```
In [132]: print(a)
```

```
3
```

numbers

integers

```
In [133]: type(2)
```

```
Out[133]: int
```

```
Out[183]: 'XPUB says Hello!'
```

```
In [184]: x.split()
```

```
Out[184]: ['XPUB', 'says', 'Hello!']
```

```
In [185]: x.startswith("x")
```

```
Out[185]: False
```

```
In [186]: x.startswith("X")
```

```
Out[186]: True
```

```
In [187]: x.find("P")
```

```
Out[187]: 1
```

```
In [188]: x.index("H")
```

```
Out[188]: 10
```

If we combine that with the f-string we used above:

```
In [17]: print(f"hello {choice(xpub1)}, welcome to my {
          choice(items)}")
hello cara, welcome to my library
```

for loop with range()

```
In [1]: for number in range(6):
          print(f"hello {choice(xpub1)}, welcome to
              my {choice(items)}")
hello cara, welcome to my studio
hello aglata, welcome to my library
hello ada, welcome to my library
hello aglata, welcome to my library
hello log, welcome to my studio
hello ada, welcome to my library
```

String operations

```
In [179]: x = "XPUB says Hello!"
In [180]: x.upper()
Out[180]: 'XPUB SAYS HELLO!'
In [181]: x.title()
Out[181]: 'Xpub Says Hello!'
In [182]: x.swapcase()
Out[182]: 'xpub says hello!'
In [183]: x.strip()
Out[183]: 'XPUB SAYS HELLO!'
In [179]: x = "XPUB says Hello!"
```

strings

```
In [137]: c = "hello"
          d = "world"
In [138]: c + d
Out[138]: 'helloworld'
In [139]: type(c)
Out[139]: str
In [140]: str(2)
Out[140]: '2'
In [141]: "3" + "2"
Out[141]: '32'
```

```
In [135]: float(2)
Out[135]: 2.0
In [136]: int(1.5)
Out[136]: 1
```

You can turn a float into an integer, sometimes this is useful. Or the other way around.

```
In [134]: type(1.5)
Out[134]: float
```

floats

It is possible to *access* specific characters in a string.

For example, you can select the first letter of a string with `[0]` or the last with `[-1]`.

```
In [142]: c[0]
```

```
Out[142]: 'h'
```

```
In [143]: c[-1]
```

```
Out[143]: 'o'
```

```
In [144]: c[1]
```

```
Out[144]: 'e'
```

lists

```
In [145]: xpub1 = ["log", "cara", "ada", "aglaia",  
                 "boyana", "suzan"]
```

```
In [146]: xpub1
```

```
Out[146]: ['log', 'cara', 'ada', 'aglaia', 'boyana',  
          'suzan']
```

```
In [147]: type(xpub1)
```

```
Out[147]: list
```

```
In [148]: xpub1[0]
```

```
Out[148]: 'log'
```

```
In [149]: xpub1[-1]
```

```
Out[149]: 'suzan'
```

```
hello suzan, welcome to my studio  
hello suzan, welcome to my classroom  
hello suzan, welcome to my library  
hello log, welcome to my studio  
hello log, welcome to my classroom  
hello log, welcome to my library  
hello cara, welcome to my studio  
hello cara, welcome to my classroom  
hello cara, welcome to my library  
hello boyana, welcome to my studio  
hello boyana, welcome to my classroom  
hello boyana, welcome to my library  
hello aglaia, welcome to my studio  
hello aglaia, welcome to my classroom  
hello aglaia, welcome to my library  
hello ada, welcome to my studio  
hello ada, welcome to my classroom  
hello ada, welcome to my library
```

pseudo-random numbers

You can load built-in *modules* or external *libraries* to use functionalities that other people wrote.

```
In [173]: from random import choice
```

```
In [174]: choice(xpub1)
```

```
Out[174]: 'suzan'
```

```
In [175]: choice(xpub1)
```

```
Out[175]: 'cara'
```

```
In [176]: choice(xpub1)
```

```
Out[176]: 'log'
```

```
In [167]: s = f"hello { name }, welcome to my { item }"
```

```
In [168]: s
```

```
Out[168]: 'hello you, welcome to my house'
```

for loops

```
In [169]: for name in xpubl:
           print(name)
```

```
suzan
```

```
log
```

```
cara
```

```
boyana
```

```
aglaia
```

```
ada
```

```
In [170]: items = ["studio", "classroom", "library"]
```

```
In [171]: for name in xpubl:
```

```
           print(f"hello { name }, welcome to my {
```

```
           items[0] }")
```

```
hello suzan, welcome to my studio
```

```
hello log, welcome to my studio
```

```
hello cara, welcome to my studio
```

```
hello boyana, welcome to my studio
```

```
hello aglaia, welcome to my studio
```

```
hello ada, welcome to my studio
```

```
In [172]: for name in xpubl:
```

```
           for item in items:
```

```
               print(f"hello { name }, welcome to my {
```

```
               item }")
```

You can also select a *range* of items in a list, which is called *slicing*:

```
In [150]: xpubl[0:3]
```

```
Out[150]: ['log', 'cara', 'ada']
```

```
In [151]: xpubl[3:]
```

```
Out[151]: ['aglaia', 'boyana', 'suzan']
```

```
In [152]: xpubl[:2]
```

```
Out[152]: ['log', 'cara']
```

You can sort a list on alphabetical order, or reversed:

```
In [153]: xpubl.sort()
```

```
In [154]: xpubl
```

```
Out[154]: ['ada', 'aglaia', 'boyana', 'cara', 'log',
           'suzan']
```

```
In [155]: xpubl.sort(reverse=True)
```

```
In [156]: xpubl
```

```
Out[156]: ['suzan', 'log', 'cara', 'boyana', 'aglaia',
           'ada']
```

string-to-list, list-to-string

string-to-list

```
In ... xpub = "XPUB focuses on the acts of making things  
public and creating publics in the age of post-  
digital networks. "
```

```
In [158]: xpub
```

```
Out[158]: 'XPUB focuses on the acts of making things  
public and creating publics in the age of  
post-digital networks. '
```

```
In [159]: xpub.split()
```

```
Out[159]: ['XPUB',  
'focuses',  
'on',  
'the',  
'acts',  
'of',  
'making',  
'things',  
'public',  
'and',  
'creating',  
'publics',  
'in',  
'the',  
'age',  
'of',  
'post-digital',  
'networks.']
```

list-to-string

```
In [160]: xpub_list = xpub.split()
```

```
In [161]: xpub_list.sort()
```

```
In [162]: xpub_list
```

```
Out[162]: ['XPUB',  
'acts',  
'age',  
'and',  
'creating',  
'focuses',  
'in',  
'making',  
'networks.',  
'of',  
'of',  
'on',  
'post-digital',  
'public',  
'publics',  
'the',  
'the',  
'things']
```

```
In [163]: " ".join(xpub_list)
```

```
Out[163]: 'XPUB acts age and creating focuses in  
making networks. of of on post-digital  
public publics the the things'
```

```
In [164]: xpub1
```

```
Out[164]: ['suzan', 'log', 'cara', 'boyana', 'aglaia',  
'ada']
```

```
In [165]: "-----".join(xpub1)
```

```
Out[165]: 'suzan-----log-----cara-----boyana-----  
aglaia-----ada'
```

f-strings

```
In [166]: name = "you"  
item = "house"
```