In [ ]:
```
print("Hello, world") # press shift+enter
```

`print()` is a function that prints the specified message to the screen.
The message can be a string, or any other object, the object will be converted into a string before written to the screen.

`"Hello, world"` is the message we want to print to the screen, it's a `string` btw

`# press shift + enter` is a comment, not a code,

Use as many useful comments as you can in your program to:

- explain assumptions
- explain important decisions
- explain important details
- explain problems you're trying to solve
- explain problems you're trying to overcome in your program, etc.

Code tells you HOW, comments should tell you WHY.

In [ ]:
```
name = input("hi, what is your name? ") # don't forget to press shift+enter
```

`name` is a variable, it stores some information

`input()` is a function that asks an input information from you

`"hi, what is your name? "` is a question program ask to you and waits your answer

what is `# don't forget to press shift+enter` ??

In [ ]:
```
city = input("where are you from? ") # ask a city name
```

In [ ]:
```
age = input("how old are you? ") # ask your age
```

In [ ]:
```
# prepare message
message = "hi " + name + ", you are from " + city + " and you " + age + " years old!"
# print message to the screen
print(message)
```

ATTENTION! here we used `+` operator to join strings together

In [ ]:
```
cheer = "Hip hip Hoorray! "
print (cheer)
```

In [ ]:
```
congratulation = cheer * 3
print(congratulation)
```

ATTENTION! here we used `*` operator to repeat string `3` times and joint together

Strings data type

Textual data in Python is handled with `str` objects, or strings. Strings are immutable sequences of Unicode code points. String literals are written in a variety of ways:

- Single quotes: 'allows embedded "double" quotes'
- Double quotes: "allows embedded 'single' quotes".
- Triple quoted: '''Three single quotes''', """Three double quotes"""

In [ ]:
```
my_string_1 = 'hello, ' + "world"
my_string_1
```

In [ ]:
```
my_string_2 = "print('hello, world') "
my_string_2
```

Triple quoted strings may span multiple lines - all associated whitespace will be included in the string literal.

In [ ]:
```
my_string_3 = '''
# print "Hello, world" message to the screen program
print("Hello, world")
'''
print(my_string_3)
```

In [ ]:
```
type(my_string_3)
# new function to show type of an object
```

In [ ]:
```
len(my_string_3)
# new function to show length of the string
```

Strings may also be created from other objects using the `str()` function.

In [ ]:
```
n = 123.456
n
```

In [ ]:
```
type(n)
```

In [ ]:
```
s = str(n)
s
```

In [ ]:
```
type(s)
```

Python object attributes (methods and properties)

Different types of objects in Python have different **attributes** that can be referred to by name (similar to a variable). To access an attribute of an object, use a dot (.) after the object, then specify the attribute (i.e. object.attribute)

When an attribute of an object is a callable, that attribute is called a **method**. It is the same as a function, only *this function* is bound to a particular object.

When an attribute of an object is not a callable, that attribute is called a **property**. It is just a piece of data about the object, that is itself another object.

The built-in dir() function can be used to return a list of an object's attributes.

In [ ]:
```
class Person:
    def hello(self): return 'hello, ' + self.name
```

In [ ]:
```
student = Person()
student
```

In [ ]:
```
student.name = 'bob'
student.name
```

In [ ]:
```
student.hello()
```

In [ ]:
```
dir(student)
```

In [ ]:
```
my_string = 'My SuPeR sTrInG'
my_string
```

- `.upper()` to return an uppercase version of the string (all chars uppercase)

In [ ]:
```
print(my_string)
print(my_string.upper())
```

- `.lower()` to return an lowercase version of the string (all chars lowercase)

In [ ]:
```
print(my_string)
print(my_string.lower())
```

- `.capitalize()` to return a capitalized version of the string (only first char uppercase)

In [ ]:
```
print(my_string)
print(my_string.capitalize())
```

- `.title()` to return a title version of the string (first char in each word uppercase)

In [ ]:
```
print(my_string)
print(my_string.title())
```

In [ ]:
```
new_string = 'annual_report.xlsx'
```

- `.count(substring)` to return the number of occurences of the substring in the string

In [ ]:
```
new_string.count('r')
```

In [ ]:
```
new_string.count('report')
```

In [ ]:
```
'ABABABABA'.count("ABA")
```

- `.replace(old, new)` to return a copy of the string with occurences of the "old" replaced by "new"

In [ ]:
```
'Happy New 2018 Year !'.replace ('2018', '2019')
```

In [ ]:
```
greeting = 'Hi, my name is Bob and I am from New-York!'
greeting
```

In [ ]:
```
greeting.replace('Bob', 'Berik').replace('New-York', 'Nur-Sultan')
```