# functions

> A function is a group of connected statements (code) that perform a specific task.

```
def function_name( input_variables ):
   something happens
   if something is True:
       then something else
   else:
       stuff here
   then, other thing happen
   maybe even another thing
   return output_variables
```

- Functions help break our code into **smaller** and **modular** parts.
- Functions make larger codes more **organized** and **manageable**
- A function does not necessarily **return** something
- When called, the content of the function is executed sequentially.

# functions

Let's write a function which tells us if today is Monday.

We will use the `datetime` package (provided with python), open your python shell:

- `date` refers to a "class" from the `datetime` package, which contains a function, `today()`, that returns today's date. Try it:

```
>>> from datetime import date
>>> date.today()
>>> datetime.date(2019, 1, 28)
```

- `datetime` also provides a function which finds the day of the week for a specific date (https://docs.python.org/3/library/datetime.html#datetime.date.weekday):

```
>>> from datetime import date
>>> today = date.today()
>>> isitmonday = today.weekday()
>>> isitmonday
>>> 0 #Because I executed this on a Monday (the 28th of January)
```

# functions

Now that we're able to determine if today is a Monday or not, let's write our function:

- Our function will take the date as an `input_variable`
- It will return a certain message if it is Monday and a different one if it isn't

```
from datetime import date

def monday_check(specimen_date):
    if specimen_date.weekday() == 0:
        message = "Monday again ... Go away Monday!"
    else:
        message = "Today is not a Monday!"
    return message
```

# functions

Let's now call our function in our script *(to execute a script* `python3 script.py`*)*:

```
#IMPORTS
from datetime import date

#FUNCTIONS
def monday_check(specimen_date):
    if specimen_date.weekday() == 0:
        message = "Monday again ... Go away Monday!"
    else:
        message = "Today is not a Monday!"
    return message

#SCRIPT
print("This program will tell you if it is already the worst day of the week.")
today = date.today()
print( monday_check(today) )
```

Try this code: Go to `examples` folder -> `ex_mondaycheck.py`

# functions : You do it now.

> In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself).
> The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6.
> Equivalently, the number 6 is equal to half the sum of all its positive divisors: ( 1 + 2 + 3 + 6 ) / 2 = 6.

- Your function should input a **number** (any integer number)
- Your function should return the **boolean value** `False` or `True`

> Hint: `message = False` affects the **boolean value** `False` to message. Not `false`, `False`. `False` is not a string, not an integer, it is a boolean.

Solution : https://www.w3resource.com/python-exercises/python-functions-exercise-11.php