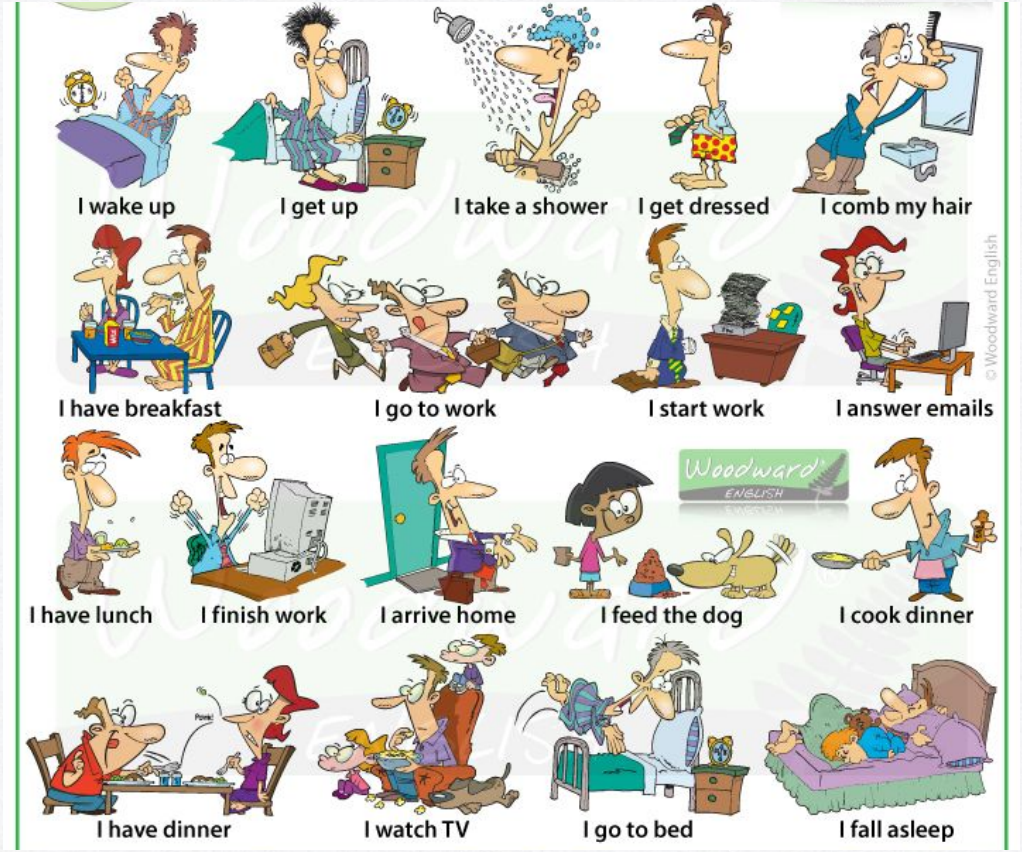
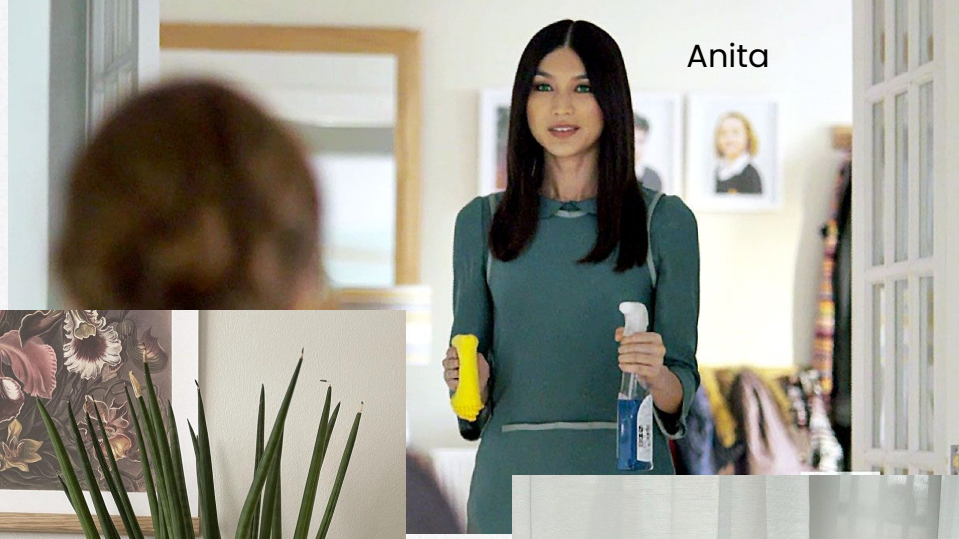
Functions in Python

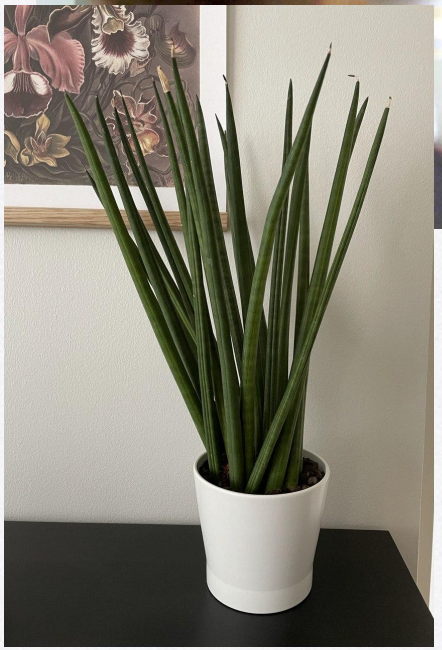
Agenda

1. Functions
2. Parameters
3. Return values
4. Things to remember

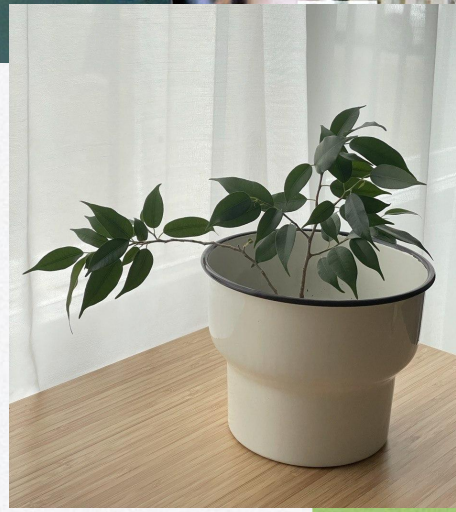

I wake up
I get up
I take a shower
I get dressed
I comb my hair
I have breakfast
I go to work
I start work
I answer emails
I have lunch
I finish work
I arrive home
I feed the dog
I cook dinner
I have dinner
I watch TV
I go to bed
I fall asleep
Woodward ENGLISH
© Woodward English

```
def main():
     for day in range(1,366):
          if day%7 == 0:
               if checkJayJay == True:
                    FillACup()
                    PourWater()
               if checkGeorge == True:
                    FillACup()
                    PourWater()



main()
```



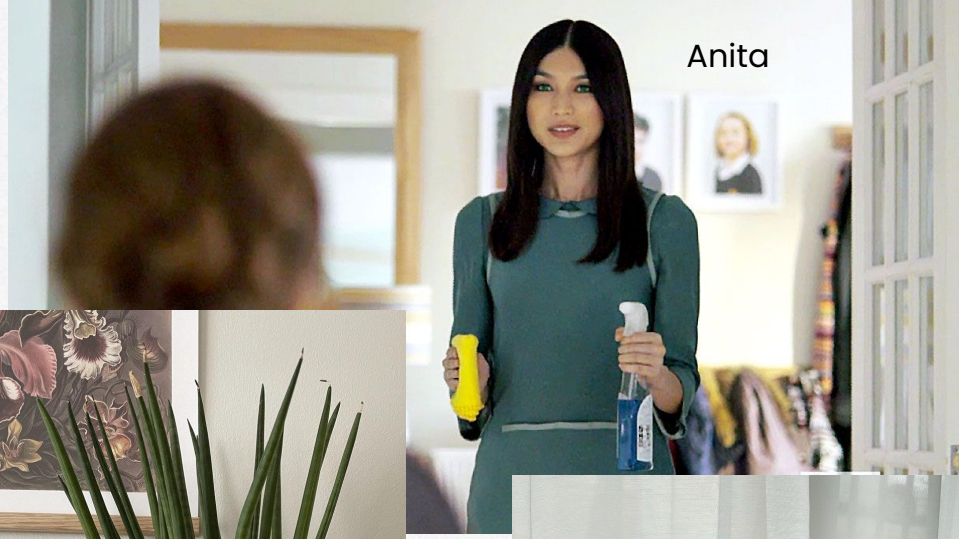


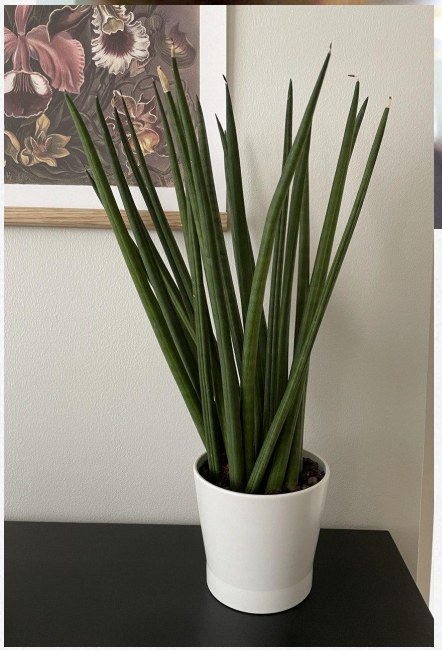
George



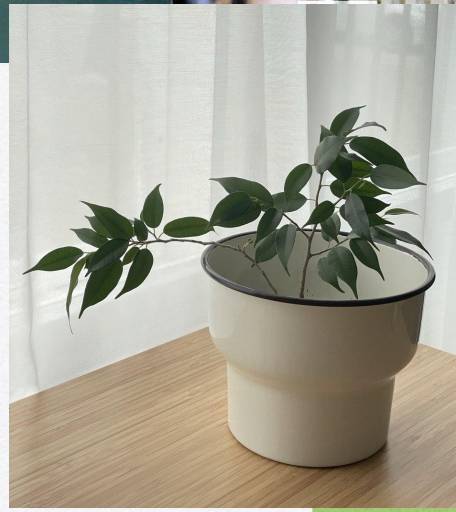
Jay-Jay

```
def main():
    for day in range(1,366):
        if day%7 == 0:
            WaterPlants()

def WaterPlants():
    if checkJayJay == True:
        Water()
    if checkGeorge == True:
        Water()

def Water():
    FillACup()
    PourWater()

main()
```





George



Jay-Jay

1

Functions

Functions

```
def buyProducts():
      buyEggs()
      buyFlour()
      buyMilk()
      buySugar()
      print("Go on now!")

buyProducts()
```

```
def buyMilk():
      cost_big = 1.2
      cost_small = 0.9
      litres_needed = 0.5
      if litres_needed <= 0.5:
            print("Buy a small pack of milk
                  for",cost_small,"eur.")
      else:
            print("Buy a big pack of milk
                  for",cost_big,"eur.")
```

```
def buyFlour():
      cost = 2.7
      print("Buy flour for",cost,"eur.")
```

```
def buyEggs():
      cost_big = 2.3
      cost_small = 1.4
      eggs_needed = 2
      if eggs_needed <= 6:
            print("Buy a small pack of eggs
                  for",cost_small,"eur.")
      else:
            print("Buy a big pack of eggs
                  for",cost_big,"eur.")
```

```
def buySugar():
      cost = 3
      print("Buy sugar for",cost,"eur.")
```

Go to:

presemo.aalto.fi/functions

What is the output?

```
def buyProducts():
      buyEggs()
      #buyFlour()
      buyMilk()
      #buySugar()
      print("Go on now!")

buyProducts()
```

```
def buyMilk():
      cost_big = 1.2
      cost_small = 0.9
      litres_needed = 0.5
      if litres_needed <= 0.5:
            print("Buy a small pack of milk
                  for",cost_small,"eur.")
      else:
            print("Buy a big pack of milk
                  for",cost_big,"eur.")
```

```
def buyFlour():
      cost = 2.7
      print("Buy flour for",cost,"eur.")
```

```
def buyEggs():
      cost_big = 2.3
      cost_small = 1.4
      eggs_needed = 2
      if eggs_needed <= 6:
            print("Buy a small pack of eggs
                  for",cost_small,"eur.")
      else:
            print("Buy a big pack of eggs
                  for",cost_big,"eur.")
```

```
def buySugar():
      cost = 3
      print("Buy sugar for",cost,"eur.")
```

2

Parameters

Parameters

```
def buyProducts():
      buyEggs(3)
      buyFlour()
      buyMilk(0.75)
      buySugar()
      print("Go on now!")

buyProducts()
```

```
def buyMilk(litres_needed):
      cost_big = 1.2
      cost_small = 0.9
      if litres_needed <= 0.5:
            print("Buy a small pack of milk
                  for",cost_small,"eur.")
      else:
            print("Buy big pack of milk
                  for",cost_big,"eur.")
```

```
def buyFlour():
      cost = 2.7
      print("Buy flour for",cost,"eur.")
```

```
def buyEggs(eggs_needed):
      cost_big = 2.3
      cost_small = 1.4
      if eggs_needed <= 6:
            print("Buy a small pack of eggs
                  for",cost_small,"eur.")
      else:
            print("Buy big pack of eggs
                  for",cost_big,"eur.")
```

```
def buySugar():
      cost = 3
      print("Buy sugar for",cost,"eur.")
```

Parameters

```
def buyProducts():
      buyEggs(3)
      buyFlour()
      buyMilk(0.75)
      buySugar()
      print("Go on now!")

buyProducts()
```

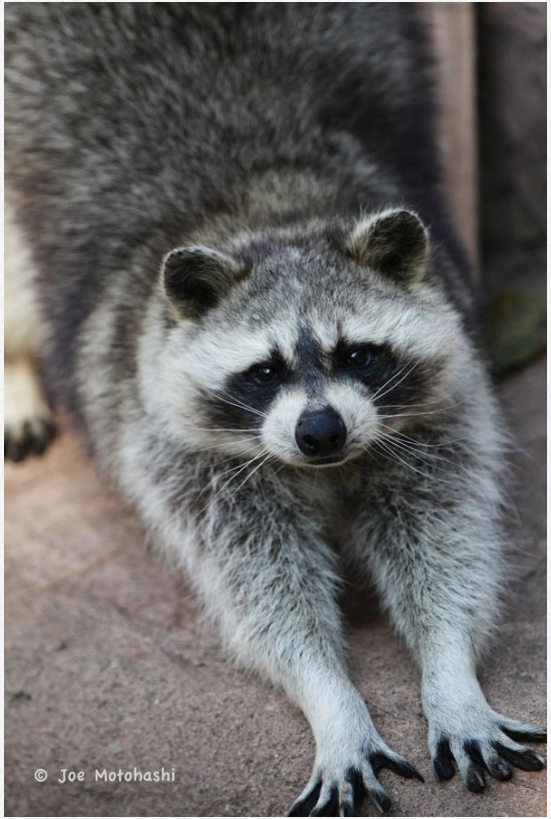
```
def buyMilk(amount):
      cost_big = 1.2
      cost_small = 0.9
      if amount <= 0.5:
            print("Buy a small pack of milk
                  for",cost_small,"eur.")
      else:
            print("Buy big pack of milk
                  for",cost_big,"eur.")
```

```
def buyFlour():
      cost = 2.7
      print("Buy flour for",cost,"eur.")
```

```
def buyEggs(amount):
      cost_big = 2.3
      cost_small = 1.4
      if amount <= 6:
            print("Buy a small pack of eggs
                  for",cost_small,"eur.")
      else:
            print("Buy big pack of eggs
                  for",cost_big,"eur.")
```

```
def buySugar():
      cost = 3
      print("Buy sugar for",cost,"eur.")
```

Stretching break!



3

Return values

Return values

```
def buyProducts():
      size, price = buyEggs(3)
      print("Buy a",size,"pack of eggs
                        for",price,"eur.")
      volume, price = buyMilk(0.75)
      print("Buy a",volume,"pack of milk
                        for",price,"eur.")
      buyFlour()
      buySugar()
      print("Go on now!")

buyProducts()
```

```
def buyMilk(amount):
      cost_big = 1.2
      cost_small = 0.9
      if amount <= 0.5:
            size="small"
            cost=cost_small
      else:
            size="big"
            cost=cost_big
      return size, cost
```

```
def buyEggs(amount):
      cost_big = 2.3
      cost_small = 1.4
      if amount <= 6:
            size="small"
            cost=cost_small
      else:
            size="big"
            cost=cost_big
      return size, cost
```

```
def buyFlour():
      cost = 2.7
      print("Buy flour for",cost,"eur.")
```

```
def buySugar():
      cost = 3
      print("Buy sugar for",cost,"eur.")
```

Can we simplify it?

```
def buyProducts():
      size, price = buyEggs(3)
      print("Buy a",size,"pack of eggs
                        for",price,"eur.")
      volume, price = buyMilk(0.75)
      print("Buy a",volume,"pack of milk
                        for",price,"eur.")
      buyFlour()
      buySugar()
      print("Go on now!")

buyProducts()
```

```
def buyMilk(amount):
      cost_big = 1.2
      cost_small = 0.9
      if amount <= 0.5:
            return "small", cost_small
      else:
            return "big", cost_big
```

```
def buyEggs(amount):
      cost_big = 2.3
      cost_small = 1.4
      if amount <= 6:
            return "small", cost_small
      else:
            return "big", cost_big
```

```
def buyFlour():
      cost = 2.7
      print("Buy flour for",cost,"eur.")
```

```
def buySugar():
      cost = 3
      print("Buy sugar for",cost,"eur.")
```

And even simpler?

```
def buyProducts():
      size, price = buyEggs(3)
      print("Buy a",size,"pack of eggs
                        for",price,"eur.")
      volume, price = buyMilk(0.75)
      print("Buy a",volume,"pack of milk
                        for",price,"eur.")
      buyFlour()
      buySugar()
      print("Go on now!")

buyProducts()
```

```
def buyMilk(amount):
      if amount <= 0.5:
            return "small", 0.9
      else:
            return "big", 1.2
```

```
def buyEggs(amount):
      if amount <= 6:
            return "small", 1.4
      else:
            return "big", 2.3
```

```
def buyFlour():
      print("Buy flour for 2.7 eur.")
```

```
def buySugar():
      print("Buy sugar for 3 eur.")
```

And as simple as that!

```
def buyProducts():
      print("Buy a {} pack of eggs
                  for {} eur.".format(buyEggs(3))
      print("Buy a {} pack of milk
                  for {} eur.".format(buyMilk(0.75))
      buyFlour()
      buySugar()
      print("Go on now!")

buyProducts()
```

```
def buyMilk(amount):
      if amount <= 0.5:
            return "small", 0.9
      else:
            return "big", 1.2
```

```
def buyEggs(amount):
      if amount <= 6:
            return "small", 1.4
      else:
            return "big", 2.3
```

```
def buyFlour():
      print("Buy flour for 2.7 eur.")
```

```
def buySugar():
      print("Buy sugar for 3 eur.")
```

Go to:

presemo.aalto.fi/functions

Which program is incorrect and why?

A

```
def travellingMode(distance,destination,bridge):
      if destination=="island" and bridge==False:
            if distance <= 30:
                  return "kayak"
            else:
                  return "nothing"
      else:
            return "bike"

def travelling():
      distance = 15
      destination = "island"
      isBridge = False
      print("You can travel to a {} by means of
                  {}.".format(destination,
                        travellingMode(distance,
                        destination, isBridge))


travelling()
```

B

```
def travellingMode(distance,destination,bridge):
      if destination=="island" and bridge==False:
            if distance <= 30:
                  mode = "kayak"
            else:
                  mode = "nothing"
      else:
            mode = "bike"
      return mode

def travelling():
      distance = 15
      destination = "island"
      isBridge = False
      mode=travellingMode(distance, destination,
                              isBridge)
      print("You can travel to a {} by means of
                  {}.".format(destination, mode))

travelling()
```

4

Things to remember

FUNCTIONS

- Let's admit, we are lazy. We want to *simplify* and *automatise* things.
- Divide and conquer: *one function - one task*!
- A variable defined in a function remains in the function (do not mix up with returning values) and does not affect a variable with the same name in another function.

PARAMETERS

- One can give no, one or multiple parameters, like this:

```
def main():
      #some code
main()
```

```
def MyFunc(some_variable):
      print(some_variable+1)
MyFunc(55)
```

```
def MyFunc(a,b):
      print(a+b)
MyFunc(5,10)
```

PARAMETERS

- The first rule of a parameter function: [almost] everything that is done to a parameter inside that function remains a secret for other functions 🤫
  In other words, parameters are "local variables", inherent only to that specific function.

```
def IDoWhatIWantWithX(x):
    x=x+10
    print(x)

def MeToo(x):
    x=x/10+5
    print(x*2)
```

This x 😁: "*I am different from you!*"*

That x 🙄: "*I know!*"

*Even though they sometimes can be equal to the same value

PARAMETERS

- Parameters passed to a function can be of different types: `integer`, `string`, `float`, `list`, etc. You can use their combinations according to your ~~taste~~ needs.

```
def cheekyFunction(coolList, fancyInteger, funkyString):
      #some code
cheekyFunction([ "c","o","o","l"], 7, "Let's party!")
```

PARAMETERS

- Parameters passed to a function can be values stored in variables or values given directly, or expressions like this:

```
def print_score(num,name):
      print("Your score is {} out of 10, {}!".format(num, name))

def main():
      score=10
      someone="Barbara"
      print_score(score,  "Visa")
      print_score(score, someone)
      print_score(9,  "Mary")
      print_score(10-2,  "John")

main()
```

RETURN VALUES

- One can return no, one or multiple values, like this:

```
def main():
      #some code
main() #traditionally, main function never returns anything
```

```
def MyFunc(some_variable):
      some_variable-=1
      return some_variable
num = MyFunc(55)
```

```
def MyFunc(a,b):
      a+=2
      b=a+b
      return a,b
x, y = MyFunc(5,10)
```

RETURN VALUES

- Values returned by a function can be of different types: `integer`, `string`, `float`, `list`, etc. You can use their combinations according to your ~~taste~~ needs.

```
def doSomething(someNumber):
      fNumber=float(someNumber)
      someNumber=100
      randomLine="Why not to return a string?"
      return fNumber, someNumber, randomLine

price,randomNum,line = doSomething(80)
```

RETURN VALUES

- Values returned by a function can be stored in variables, given directly, or as a result of an expression like this:

```
def MyFunc(x):
      x=x**2
      return x, "Mind the gap!", x*0.1
num, line, anotherNum = MyFunc(5)
```

RETURN VALUES

- Remember, values returned by a function have to be used somehow! Otherwise they will "hang in the air" (not to say it's *bad coding*).
  Either save them in variables (remember to use as many of variables as there are returned values) or print them out, or use in any other way.

```
def doSomething(someNumber):
      fNumber=float(someNumber)
      randomLine="Why not to return a string?"
      return fNumber, randomLine

price,line = doSomething(80)
#OR you could do this
print(doSomething(80))
#but not this!
doSomething(80)
```

RETURN VALUES

- A function must return values only once! However, you still can set conditions when to return which value, like this:

```
def MyFunc(x):
      if x>10:
            return "It's bigger than 10!"
      else:
            return "It's smaller than 10… Or it is 10."
print(MyFunc(5))
```

RETURN VALUES

- Remember not to leave condition branches 'hanging' without `return`-s like this:

```
# DO NOT DO LIKE THIS!
def MyFunc(x):
      if x>10:
            return "It's bigger than 10!"

line = MyFunc(5)
```

- Also remember that when an interpreter meets `return` in the function, it's going to quit it at that spot whether you want it or not.

Fin.