When you write the name of a variable for the first time, this is called to declare a variable. When you give the variable a value, this is called assign a value to a variable. This is done with the = sign.

Examples of variable declaration and assignment:
name = “Hermione”
age = 7
family = [“James”, “Martha”, “Walter”, “Loulou”, “Leila”]

#Task 1:

Find and colour all the variables, only once, when they appear the first time (when they are declared and assigned their first value).

How many different variables are there in “guess the number”?

```
# guess the number
import random

guessesTaken = 0

print('Hello! What is your name?')
myName = input()

number = random.randint(1, 20)
print('Well, ' +myName+ ', I am thinking of a number between 1 and 20.')

while guessesTaken < 6:
    print('Take a guess.')
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')

    if guess > number:
        print('Your guess is too high.')

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print('Good job, ' +myName+ '! You guessed my number in ' +guessesTaken+ ' guesses!')

if guess != number:
    number = str(number)
    print('Nope. The number I was thinking of was ' +number)
```

A variable can be used again and again.
It is also possible to change the value of a variable. If you use it after you changed its value, it has the new value.

Examples of changing the value of variables:
name = “Hermione Mae”
age = 7¾
family = [“James”, “Martha”, “Walter”, “Loulou”, “Leila”, “Joe”, “Ida”, “Lalo”]

#Task 2:

Highlight or colour each variable (and all its occurrences) in its own colour.

Is there a variable that changes its value? Can you underline it?

```
# guess the number
import random

guessesTaken = 0

print('Hello! What is your name?')
myName = input()

number = random.randint(1, 20)
print('Well, ' +myName+ ', I am thinking of a number between 1 and 20.')

while guessesTaken < 6:
    print('Take a guess.')
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')

    if guess > number:
        print('Your guess is too high.')

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print('Good job, ' +myName+ '! You guessed my number in ' +guessesTaken+ ' guesses!')

if guess != number:
    number = str(number)
    print('Nope. The number I was thinking of was ' +number)
```

Functions are always doing something. They are the action in a programme.
They have a name and after the name a bracket. The brackets can be empty or have something in them.
If there is something inside the brackets, that's called the arguments or the parameters of the function.

Examples of functions:
print("hello Hermione")
input()

#Task 3:

Find and colour all the functions in the "guess the number" programme. Colour the name, the brackets and everything inside the brackets, if they are not empty.

```
# guess the number
import random

guessesTaken = 0

print('Hello! What is your name?')
myName = input()

number = random.randint(1, 20)
print('Well, ' +myName+ ', I am thinking of a number between 1 and 20.')

while guessesTaken < 6:
    print('Take a guess.')
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')

    if guess > number:
        print('Your guess is too high.')

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print('Good job, ' +myName+ '! You guessed my number in ' +guessesTaken+ ' guesses!')

if guess != number:
    number = str(number)
    print('Nope. The number I was thinking of was ' +number)
```

Sometimes functions are called methods. They are called methods, when they belong to somebody (the owner). Only the owner can use the methods that belong to him/her.
A method starts with the owner, followed by a dot, followed by the method and of course the brackets.

#Task 4:

There is only one method in your “guess the number” programme. Can you find it?

Can you write down the name of the owner, then name of the method and the value of the argument?
Hint: Look for this pattern: owner.methodname(argument). Look out for the dot and the brackets at the end of the method name.

```
# guess the number
import random

guessesTaken = 0

print('Hello! What is your name?')
myName = input()

number = random.randint(1, 20)
print('Well, ' +myName+ ', I am thinking of a number between 1 and 20.')

while guessesTaken < 6:
    print('Take a guess.')
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')

    if guess > number:
        print('Your guess is too high.')

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print('Good job, ' +myName+ '! You guessed my number in ' +guessesTaken+ ' guesses!')

if guess != number:
    number = str(number)
    print('Nope. The number I was thinking of was ' +number)
```

There are different types of variables. You already now Strings, Numbers and Lists. (In other programming languages, lists are called arrays.) Booleans are another type: There are only 2 possibilities: True or False.

Weird: In programming “equal” is written as == (4==4 or “Hermione”==“Hermione”)
A single = is an assignment, giving a value to a variable (name = “Hermione”, hair = “blonde”)

== equal
!= not equal
< left side smaller than right side (4 < 5)
> left side is bigger than right side (5 > 4)

#Task 5:

Colour every expression that can only be True of False.
Hint: Look for statements starting with “if” or “while”

```
# guess the number
import random

guessesTaken = 0

print('Hello! What is your name?')
myName = input()

number = random.randint(1, 20)
print('Well, ' +myName+ ', I am thinking of a number between 1 and 20.')

while guessesTaken < 6:
    print('Take a guess.')
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')

    if guess > number:
        print('Your guess is too high.')

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print('Good job, ' +myName+ '! You guessed my number in ' +guessesTaken+ ' guesses!')

if guess != number:
    number = str(number)
    print('Nope. The number I was thinking of was ' +number)
```

Blocks are lines of code that are run (executed) together. Blocks in Python are indented 4 spaces from the code they belong to. There can be blocks in blocks in blocks in blocks…

Example of a block:

if name == “Hermione”:
    hair = “blonde”
    eyes = “blue”

if the variable name is “Hermione”, then both variables in this block will be assigned: hair will be blonde AND eyes will be blue: The whole block is run (or executed) together.

#Task 6:

Colour all the blocks in one colour.

Then underline the blocks inside blocks in a different colour.

```
# guess the number
import random

guessesTaken = 0

print('Hello! What is your name?')
myName = input()

number = random.randint(1, 20)
print('Well, ' +myName+ ', I am thinking of a number between 1 and 20.')

while guessesTaken < 6:
    print('Take a guess.')
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')

    if guess > number:
        print('Your guess is too high.')

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print('Good job, ' +myName+ '! You guessed my number in ' +guessesTaken+ ' guesses!')

if guess != number:
    number = str(number)
    print('Nope. The number I was thinking of was ' +number)
```

Different types of variables can do different things. For example, if you + strings or + numbers, this will have a different result.

Example of string + (string concatenation or cat):

name = “Hermione” + “ “ + “Mae” + “ “ + “Lovelace”
print(name) >> Hermione Mae Lovelace
print(“My name is: “ + name) >> My name is Hermione Mae Lovelace

Example of number + (addition):

age = 3 + 2 + 2
print(age) >> 7

#Task 7:

Colour all the string + (string cats) in one colour and the number + (number additions) in a different colour.
What do you think will be the result of print(“3”+”2”+”2”). Try it in the Terminal.

```
# guess the number
import random

guessesTaken = 0

print('Hello! What is your name?')
myName = input()

number = random.randint(1, 20)
print('Well, ' +myName+ ', I am thinking of a number between 1 and 20.')

while guessesTaken < 6:
    print('Take a guess.')
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')

    if guess > number:
        print('Your guess is too high.')

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print('Good job, ' +myName+ '! You guessed my number in ' +guessesTaken+ ' guesses!')

if guess != number:
    number = str(number)
    print('Nope. The number I was thinking of was ' +number)
```

Changing type: Sometimes it is necessary to change the type of a variable (type conversion), so you can do stuff with it.

Example:

age = 7 (7 is a whole number, called integer)
print(“Hermione is “ +age+ “ years old.”) >> TypeError: Can't convert 'int' to str implicitly
print(“Hermione is “ +str(age)+ “ years old.”) >> “Hermione is 7 years old”

If you concatenate strings, all the bits have to be strings. You have to change your number 7 into a string “7” to use it in a string concatenation. (The function str() does this.)

#Task 8:

Colour the function that changes a string into a number in blue.

Then colour the functions that change a number into a string in red.

Why do you think guess = int(guess) is necessary? This function changes the variable guess from a string into a whole number (integer).

```
# guess the number
import random

guessesTaken = 0

print('Hello! What is your name?')
myName = input()

number = random.randint(1, 20)
print('Well, ' +myName+ ', I am thinking of a number between 1 and 20.')

while guessesTaken < 6:
    print('Take a guess.')
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')

    if guess > number:
        print('Your guess is too high.')

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print('Good job, ' +myName+ '! You guessed my number in ' +guessesTaken+ ' guesses!')

if guess != number:
    number = str(number)
    print('Nope. The number I was thinking of was ' +number)
```

Python comes with a lot of functions ready to use. You already know print() or input(). These are functions that are built into the language.
Other things are also part of the language, but they are kept in separate libraries. If everything was in the same library, it would be very big. It is better to split things up into smaller libraries, called modules.
If you need to use functions that are not part of the built in library, you have to import them. Random is an example of that. If you need to make random numbers, you need to tell python to import the random module.

import random

#Task 9:

Why do you think the import statement is the very first statement?

Play around with the random.randint() method. Try random.randint(1,20) several times. What happens?

Try random.randint(1,4) or random.randint(1,2000).
What do you expect to happen?

```
# guess the number
import random

guessesTaken = 0

print('Hello! What is your name?')
myName = input()

number = random.randint(1, 20)
print('Well, ' +myName+ ', I am thinking of a number between 1 and 20.')

while guessesTaken < 6:
    print('Take a guess.')
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')

    if guess > number:
        print('Your guess is too high.')

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print('Good job, ' +myName+ '! You guessed my number in ' +guessesTaken+ ' guesses!')

if guess != number:
    number = str(number)
    print('Nope. The number I was thinking of was ' +number)
```

Loops: The name says it all. Loops loop. They run their blocks as many times as you tell them to do so. They are brilliant at repeating tasks. In your “guess the number” game you are using a while loop. It repeats 6 times, until the guessesTaken variable is 6. Then the programme leaves the loop and moves on to the first line that is outside the block (not indented anymore).

If your guesser guesses the right number before your loop has looped 6 times, your programme needs to jump out of the loop early. You can jump out of a loop early with a break statement.

#Task 10:

Draw a red frame around the while loop.

When the while loop is finished, on which line will the programme go on?

If you only allow 3 guesses in your game, what do you need to change in your while statement?

If your guesser guesses the answer before he has used up all his guesses, your programme needs to jump out of the loop. Can you find and colour the statement that stops the loop before it has run 6 times?

```
# guess the number
import random

guessesTaken = 0

print('Hello! What is your name?')
myName = input()

number = random.randint(1, 20)
print('Well, ' +myName+ ', I am thinking of a number between 1 and 20.')

while guessesTaken < 6:
    print('Take a guess.')
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')

    if guess > number:
        print('Your guess is too high.')

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print('Good job, ' +myName+ '! You guessed my number in ' +guessesTaken+ ' guesses!')

if guess != number:
    number = str(number)
    print('Nope. The number I was thinking of was ' +number)
```