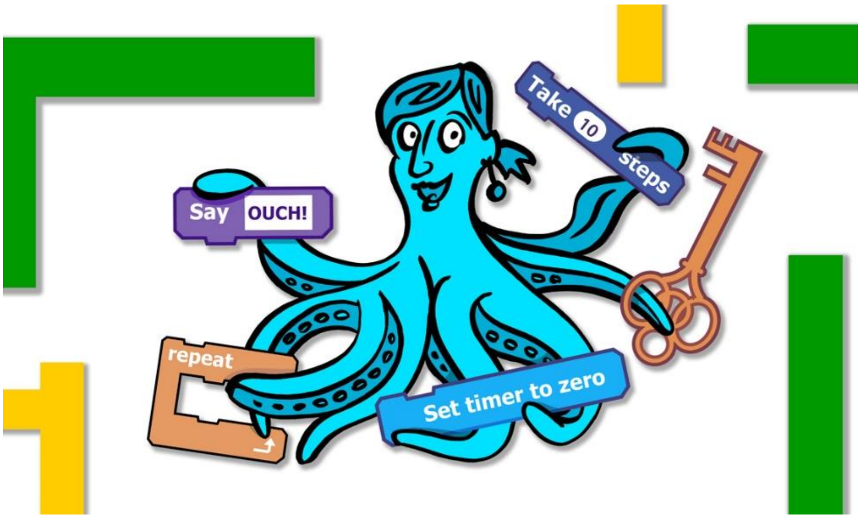
Course material – Class 8 – Python

Scratch: Programming for Teachers

MOOC – starts October 8th, 2018



This content has been created by:
Dr. Ir. Felienne Hermans

Week 8 - Python

Beforehand

Today, your class will be programming, maybe even for the first time! That's great. Our research showed that it's good for their acknowledgements of kids that they, before they will work on the computer, start with doing the assignment without the computer. So, to practice the exercise away from the computer. Do this in the beginning of your class. You can use these assignments, and of course you can re-arrange them in the way you like.

What can you all do and make with a computer?

It is nice to let your students think about what they can create with a computer. Let the kids write these things down. Quickly they'll come up with things like apps, websites and games. That is why it is fun for them to use the computer to make an interactive story! Do they come up with that as well? Guide them into the right direction.

Lists

This class, we will be working with lists. Lists are important for computers to store things in. But, these lists also occur in the "free nature".. What do your students know about lists? Which lists can they come up with? A nice example is the class list, with all the names of your students on, but there are many more!

What things can they come up with to put on a list? You ask the student one by one, or just in class, or make them write this down again.

Mostly they come up with many diverse things! Lists of friends, hobbies, favorite foods, things in class etc.

You can extend this exercise a bit. For lists they come up with: what is and what is not in there? What's also fun to do is to ask them to think about what lists are changing and what lists are not. Your friends for example, do change, but not a lot. Your brothers and sisters never change, but it could be that it changes when you get a new brother or sister… Challenge them, let them associate all these things and teach them a deep understanding of what a list is. That is central in this programming assignment in class.

Looking back

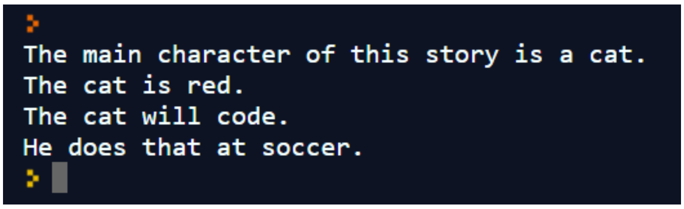
After class, you can still do something with the stories you came up with, if you want, for example let the kids re-draw their stories! Or, let them reflect on the differences in each story.

Make your own interactive story with Python

Computers can also talk!

Computers can do anything, even story telling. But, you do have to tell that to your computer, so you have to program that. Today, we're going to do that together! We will feed the computer some lists and from there, the computer picks a random option. Just as your teacher does that by picking a random kid in class to ask whether he or she wants to do a little task for him/her.

We make lists, with animals, places, activities and colors and then we place them all together in a sentence. You will get really strange sentences! "Does your green cat walks through the forest?" or "Does your lizard dig at school?". Let's have a look at it! At the end of the class, you'll be having a story like this:



```
›
The main character of this story is a cat.
The cat is red.
The cat will code.
He does that at soccer.
›
```

Or, maybe you come up with something totally different.

A few tips

- Computers are very precise, read carefully!
- Don't copy code, but type it yourself by hand
- Don't forget to press the 'start' button occasionally to see whether everything works
- Everytime you see the word **assignment** you need to type something somewhere

Have fun!

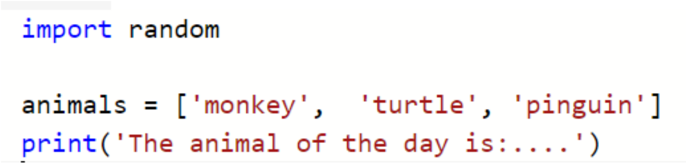
Your first Python code

Today, you will be working with Python for the first time. Python is a very well known programming language. Maybe you've never heard of it. But, you have worked with Python, as YouTube is made in Python.

Okay, we won't be programming that far today… But, we will work a lot in Python today.

To start, you can go to repl.it → https://repl.it/@sabinev/StoryStart

In there, a little program is ready for you.



```
import random

animals = ['monkey',  'turtle', 'pinguin']
print('The animal of the day is:....')
```

For now, you can forget about the “import random”. That is to tell Python that it has to do something later with random numbers. The next line is important, this creates a new list namely.

```
animals = ['monkey',  'turtle', 'pinguin']
```

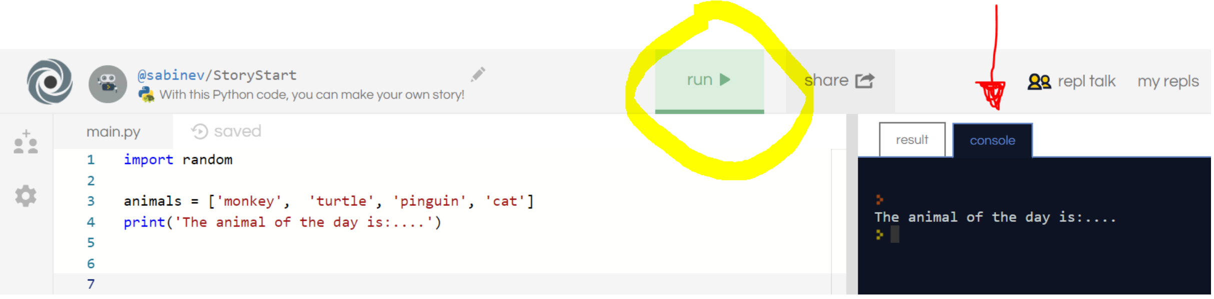
This list contains 3 animals. Just as the list you practiced in class, like the class list.

Assignment: If you want, you can add another animal. Be aware that you place a comma between the animals as well as the quote sign ‘ ‘ . Otherwise, Python will get confused!

Example:

```
animals = ['monkey',  'turtle', 'pinguin', 'cat']
```

The third line makes the code really fun, as this line “prints” something on the screen. Press the ‘start’ button as highlighted yellow in the image below:



```
import random

animals = ['monkey',  'turtle', 'pinguin', 'cat']
print('The animal of the day is:....')
```

```
The animal of the day is:....
```

Can’t see it? Make sure you clicked “console”, you can find that at the red arrow.

Assignment: make something else from the current print, for example “The main character of this story is:…”. But you can come up with anything you like!

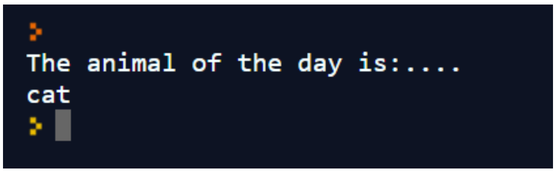
Let’s now pick an animal from the list, to play our main character in the story. We will do that with the following Python code:

```
main  = random.choice(animals)
print(main)
```

Copy this code exactly how it’s written in the above, by hand! That’s very important. A computer can’t easily read a typo and fix it. It needs very clear (and exact) instructions.

Now, press the start button. What is happening? Write it down below:

Python printed two things! First, the first sentence, followed by an animal. That animal is saved now in a variable (let the kids repeat this word: “va-ri-a-ble”.) This variable is the main.character!



```
>
The animal of the day is:....
cat
>
```

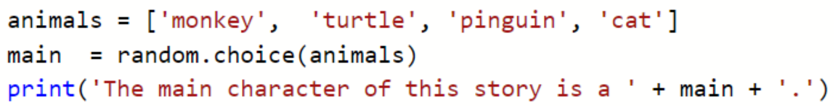
Assignment: press the start button again and run the program another time. What is happening now?

You get a different animal from the list!

Assignment: if you can’t find a good animal, you can add more animals to it, or get rid of some of the animals. Pay close attention, that between the animals there’s a comma (,), and that the animals are placed in between the quotes ‘ ‘ , otherwise Python will get confused.

A better sentence

This works, but it doesn’t look really like a story you read in a book. So, we will make it a little bit prettier by sticking parts of the sentences together. We will do this with the “+” sign. You can erase the sentences we had with ‘print’ , and let’s make it like this:



```
animals = ['monkey',  'turtle', 'pinguin', 'cat']
main  = random.choice(animals)
print('The main character of this story is a ' + main + '.')
```

Now, we are really getting a nice, pretty sentence! Try it!

[Until so far, the class for Masters] .

Another sentence

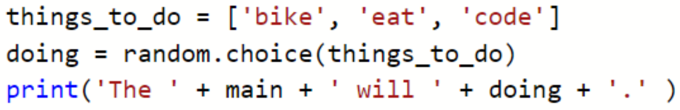
That’s going well, isn’t it? You will be getting another animal every time in new sentence. But, a good story has multiple sentences, right? So let’s add another sentence to it.

We will be doing the same things, and this consists of the following three:

1) We make a list with [‘ ‘ ,‘ ‘ , ‘ ‘]
2) We retrieve something from the list with random.choice()
3) We print it on the screen with print()

We will do something and follow the three steps above, however, we won’t be working with animals, but with *things.*

Assignment: Write new code with *things* the same way, and write it underneath your animals-code.

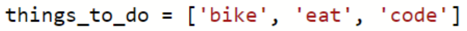


```
things_to_do = ['bike', 'eat', 'code']
doing = random.choice(things_to_do)
print('The ' + main + ' will ' + doing + '.' )
```

Try out the program! What is happening now? Yes! We will be getting 2 sentences! It starts to look like something. Press again on the ‘start’ button. Do you like the story?

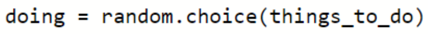
Try and come up with some other animals and things to do, if you want.

Assignment: will you write down in your own words what will be happening, after every sentence, in your own words?



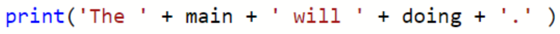
```
things_to_do = ['bike', 'eat', 'code']
```

____



```
doing = random.choice(things_to_do)
```

____



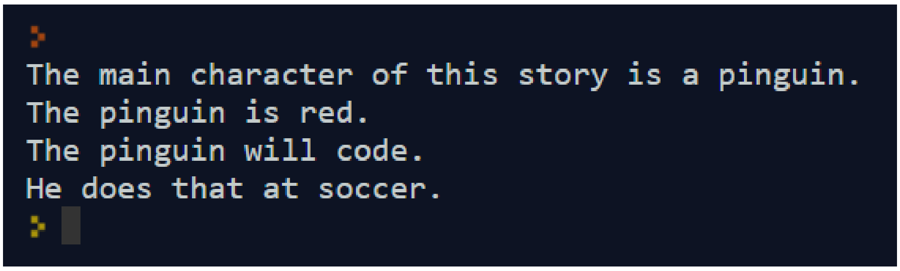
```
print('The ' + main + ' will ' + doing + '.' )
```

____

There we go, that works well, right? Let's do this another time, but this time, you can do it yourself, I think!

Will you add colors and places? Continue with making lists, pick a random item out of it and print it on the screen.

I did it like this:



```
The main character of this story is a pinguin.
The pinguin is red.
The pinguin will code.
He does that at soccer.
```

Together with a friend!

If you have any time lest, you can also program a friend with it. For that, we will be adding another animal, and place this all the way at the bottom of your program you already have:

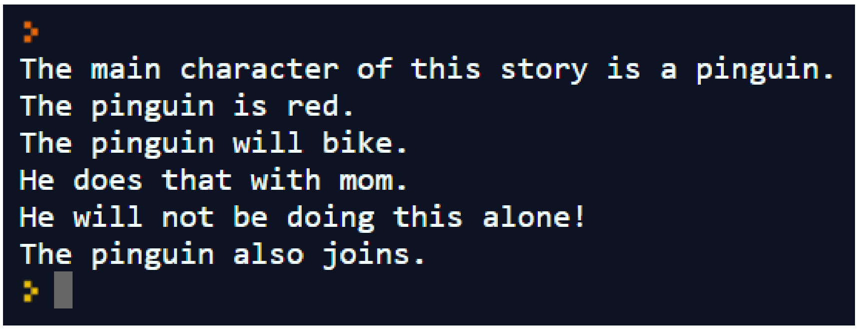
```
print('He will not be doing this alone!')
friend  = random.choice(animals)
print('The ' + friend + ' also joins.')
```

Double animals

We're getting a real story now! Only… something is not going perfectly well, now is it? What is going wrong?

0. You always get the same story
0. Something you get the same animal twice

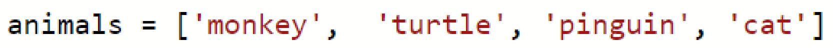
Exactly! We can get the same animals twice, for example like this:



```
The main character of this story is a pinguin.
The pinguin is red.
The pinguin will bike.
He does that with mom.
He will not be doing this alone!
The pinguin also joins.
```

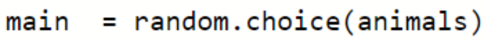
That’s a little odd. It would be more fun should we get two different animals. For that, we need to learn a new programming concept, namely that we have to erase something out of a list.

We learned how to put something in a list, right? That goes like this:



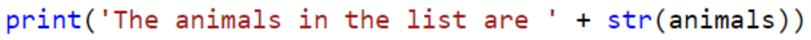
```
animals = ['monkey',  'turtle', 'pinguin', 'cat']
```

Then we randomly chose one animal from the list, which goes like this:



```
main  = random.choice(animals)
```

By randomly choosing one animal, doesn’t mean it’s not in the list anymore. It is still part of the list. You can check what is all in the list by printing it like this:

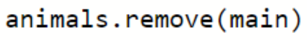


```
print('The animals in the list are ' + str(animals))
```

Place this on line 5, you will notice that random.choice will not pick an animal from the list.

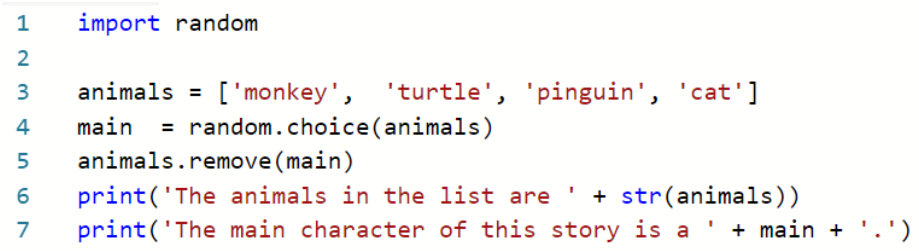
All animals are still there. So, if we want to pick another animal, changes are we get the same animal…

Place this on line 5 now:



```
animals.remove(main)
```

Your program starts like this:



```
1   import random
2
3   animals = ['monkey',  'turtle', 'pinguin', 'cat']
4   main  = random.choice(animals)
5   animals.remove(main)
6   print('The animals in the list are ' + str(animals))
7   print('The main character of this story is a ' + main + '.')
```

Look! Now the main character will be removed from the list. Try it in your program a few times. No double animals occur!

Make it interactive

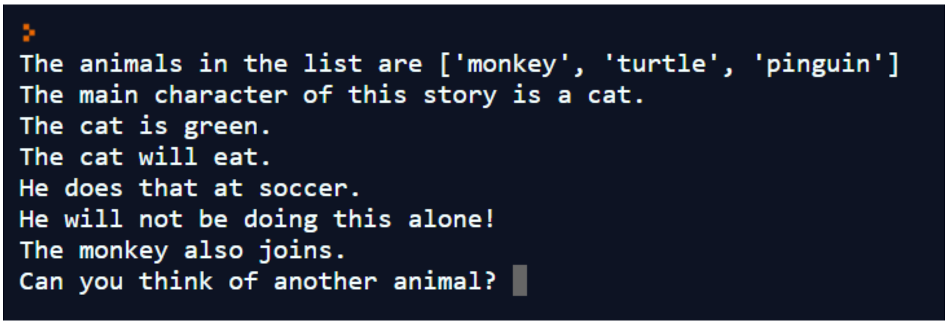
It will become even nicer should you be able to type something yourself in the story, to make it a bit different! We can do this with the function *input.*

Assignment: Add the following sentence, all the way at the bottom of your program.

```
another_animal  = input('Can you think of another animal?')
```

And press the start button again.

Now, you can type in something in the black (console) screen:



```
The animals in the list are ['monkey', 'turtle', 'pinguin']
The main character of this story is a cat.
The cat is green.
The cat will eat.
He does that at soccer.
He will not be doing this alone!
The monkey also joins.
Can you think of another animal?
```

Try to fill in an animal! For example a snake. That's fun, because Python is of course a snake 😉

What is happening now…? Nothing yet! We still have to do something with the animal, which is stored in the variable 'another_animal'.

What do you think we should use?

0. Print
0. random.coiche
0. remove

Exactly! A print.

Assignment: Print the animal in a nice sentence!

[end of class]