

PROJECT: GETTING STARTED WITH PYTHON

OAKWOOD FLL



Python is a powerful and popular programming language

Python is the third most popular programming language used today (after Java and C). It is both powerful enough to solve problems like we need to solve, and relatively easy to learn.

repl.it is a website where we can play with Python in a web browser

We will start off developing Python code at the website repl.it. Open this in the browser, and then select "+ New Repl" at the upper right. Then click on "Python" and click the button that says "Create Repl".

In the middle of the screen, there is an editor that says "main.py" at the top where we can write entire programs. The black, right side of the screen is an interpreter where we can type a line of code to see immediately what will happen. (Also, when we run a program, the output will appear on the right side of the screen.)

For example, click on the right side so the cursor blinks there, and type 1+1 and hit enter. What happens?

Writing our first Python program

Now, let's look at our first Python program:

```
name = "Mr. Lyle"  
print("Hello" + name)
```

What do you think will happen when we type in this program and run it?

To run it, type it in the middle ("main.py") window. **Be careful**, because text programming languages are very picky about punctuation and capitalization. Then, click the green "Run" program at the top of the screen.

What happened? Did anything go wrong? How can we fix our program?

Loops are useful

Usually when we write a program, there are things we want to do many times. We can use a loop to accomplish this.

```
print("I am starting")  
for i in range(5):  
    print("My number is", i)  
    print("I am all done")
```

Note that the first line has a colon at the end and that some lines are indented. Sometimes the editor will indent for us, but if not, we can indent text in our editor with the **tab** key. We can use backspace to remove indentation we don't want. Python uses indentation to know whether a line of code is in a loop and should be run more than once.

What do we expect this program will do? Type it into the middle window and run it. Did we get what we expected? How can we fix it? What would we change if we want to start counting at 1?

Lists

```
names = [ "Mr. Lyle", "Mrs. Lyle", "Mr. Ota" ]  
for name in names:  
    print("One coach is", name)
```

Nesting

We can put loops inside loops. For example:

```
names = [ "Mr. Lyle", "Mrs. Lyle", "Mr. Ota" ]
for name in names:
    print("Hi,", name, "watch me count!")
    for i in range(3):
        print(i)
```

What happens?

Or for a more complicated example:

```
print("I am starting")
for i in range(5):
    print("I am going to count to", i)
    for j in range(i):
        print(j)
    print("Done!")
```

What happens? How can we make this better? Play with it and try to improve it!

Modules

Python includes modules that can perform special tasks for us, like knowing what time it is or writing to files. To use a module, we need to **import** it first.

```
import time
print("The higher I count, the slower I go!")

for i in range(5):
    print(i)
    time.sleep(i)

print("All done!")
```

There is documentation on all the modules that come with Python at <https://docs.python.org/3.8/library/>

Functions

We can incorporate code that we will use often into a function. Another way to write the counting example would be like this:

```
def count_to(number):
    print("I will count to", number)
    for i in range(number):
        print(i)

for i in range(4):
    count_to(i)
```

It may be easier to fix the problems in this version than the earlier version...

Variables

Sometimes we have used `i`, `j`, `names`, etc. These are variables: they hold onto something (a string, list, number, or other type) until we are ready to use it in our code.