Lesson 2: Introduction To Variables

Variables

Sometimes when writing programs, you'll want to store data. There may be many reasons for this, like taking in input, organizing data, using information that can be changed later. This data is stored in a variable.

```
age = 17
print("Hi, I'm Peter. I'm {} years old.".format(age))
```

1. We store the integer value 17 in the *age* **variable**.
2. We then use the **format()** method to format our string with that variable - more on this later

If we run the program we just wrote, we'll find that our string prints out with the number 17 in place of the brackets.

```
pschwendy@Peters-MacBook-Pro Lesson 2 % python age.py
Hi, I’m Peter. I’m 17 years old.
```

We can also store variables of different types and information.

```
name = "Peter"
age = 17
print("Hi, I'm {}. I'm {} years old.".format(name, age))
```

1. We use the variable *name* to store the string “Peter”
2. We pass in variables to the **format** method to replace the brackets in order

The program outputs exactly the same as before.

```
pschwendy@Peters-MacBook-Pro Lesson 2 % python name_and_age.py
Hi, I’m Peter. I’m 17 years old.
```

Changing Variables

In python, variables are mutable by nature. That means that you can change them later on in the program.

```
age = 17
print("I'm {} years old.".format(age))
age = 20
print("I'm {} years old.".format(age))
```

```
pschwendy@Peters-MacBook-Pro Lesson 2 % python3 mutable_age.py
I'm 17 years old.
I'm 20 years old.
pschwendy@Peters-MacBook-Pro Lesson 2 %
```

1. We change the variable *age* to store the value 20
2. The print statement outputs different information base on the updated value of *age*

Variable Data Types

In many languages, you have to manually assign data types to variables. In python, types are inferred, but it's still good to know about them. We've already seen integers and strings, but there are some others to know about.

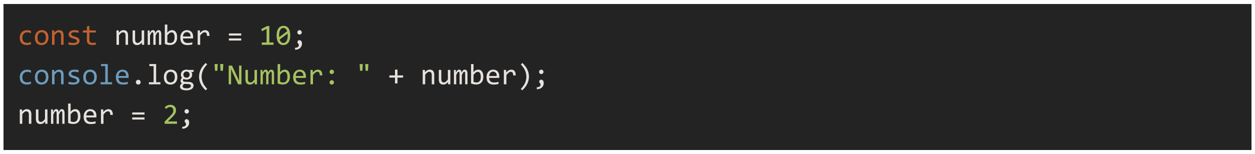
- Numbers
    - Integers (whole numbers, aka -1000, -1, 0, 1, 2, 3…, no decimal places)
    - Floating Point Numbers (non-whole numbers)
    - Complex Numbers
- String (text)
- Boolean (True or False)

Here's some way to assign those data types in code:

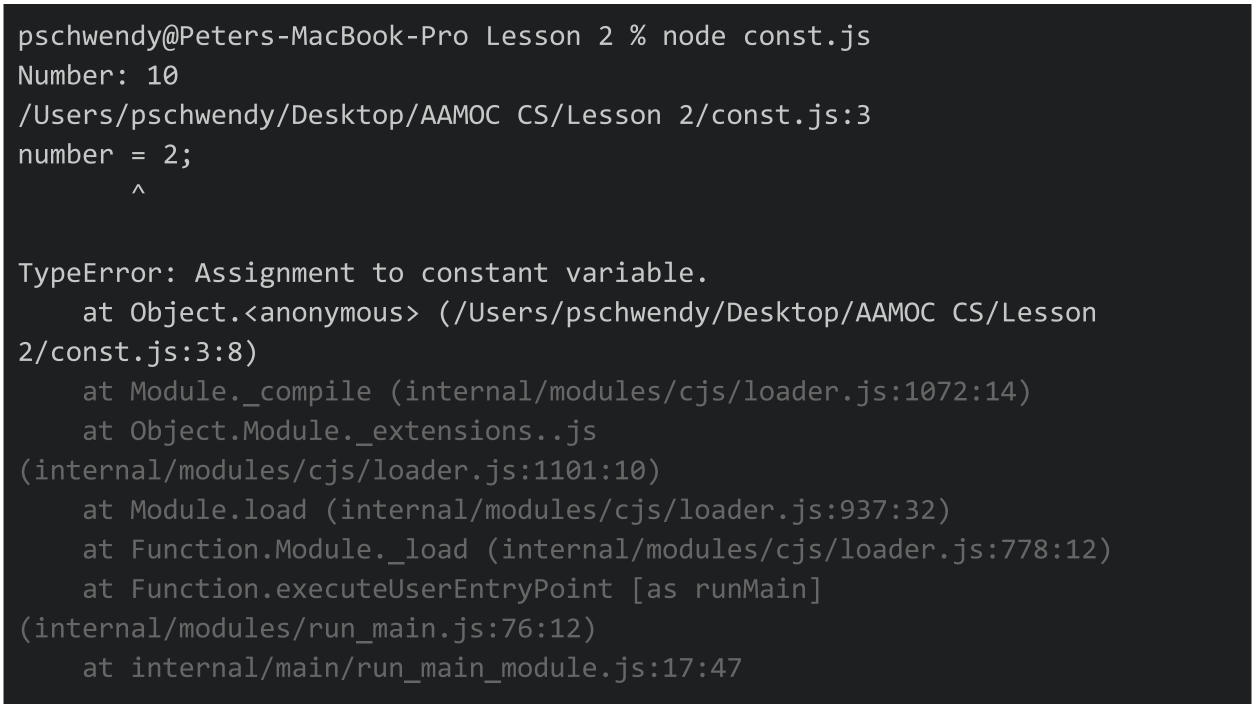
```
age = 17 # integer
pi = 3.1415926535 # float
complex = 3.14j # complex number (j = i)
name = "Pranav" # string
condition = True # bool
```

Constants

In many programming languages, variables can be declared constant with the const keyword. These types of variables cannot be changed later in the program. Let’s look at some javascript:



```
const number = 10;
console.log("Number: " + number);
number = 2;
```



```
pschwendy@Peters-MacBook-Pro Lesson 2 % node const.js
Number: 10
/Users/pschwendy/Desktop/AAMOC CS/Lesson 2/const.js:3
number = 2;
       ^

TypeError: Assignment to constant variable.
    at Object.<anonymous> (/Users/pschwendy/Desktop/AAMOC CS/Lesson
2/const.js:3:8)
    at Module._compile (internal/modules/cjs/loader.js:1072:14)
    at Object.Module._extensions..js
(internal/modules/cjs/loader.js:1101:10)
    at Module.load (internal/modules/cjs/loader.js:937:32)
    at Function.Module._load (internal/modules/cjs/loader.js:778:12)
    at Function.executeUserEntryPoint [as runMain]
(internal/modules/run_main.js:76:12)
    at internal/main/run_main_module.js:17:47
```

As you can see, the javascript interpreter throws an error when I try to change the constant *number*. Constants don’t exist in python, so technically all variables are mutable, but knowledge of constants may come in handy later.

Summary

1. Variables can store different types of data
    1. We can use the ***format*** method to put variables into strings
2. In python, there are three main data types
    1. Numbers
        1. Integer
        2. Floating point
        3. Complex
    2. Strings
    3. Booleans
3. In python, variables are mutable by nature, meaning you can change them in different parts of the program.
4. Some other languages have constant variables, which are immutable and cannot be changed