

CSP1150/CSP5110: Programming Principles

Reading 1.1: Constants

In this week's lecture we covered variables - the ability to assign a name to data stored in memory, so that you can refer to the data later. You can typically assign a new value to a variable at any time:

```
value = 25
print(value) # display content of value variable - 25

value = 'Potato'
print(value) # display content of value variable - Potato
```

Python

One thing that was only mentioned in a slide note is the concept of a *constant*. Simply put, a constant is a variable that you *cannot change the value of* - you define its value when you create the constant, and cannot assign a new value to it after that point.

Most languages support constants, and the convention is to give them UPPER_CASE names. Here are some examples of code to define a constant in various languages.

```
define('MAX_SIZE', 100);
```

PHP

```
static final int MAX_SIZE = 100;
```

Java

```
static const int MAX_SIZE = 100;
```

C / C++

Why are they useful?

Constants are used to define *values that need to be used multiple times in a program*. Rather than needing to specify the value itself each time, you just specify the name of the constant.

This makes it much easier to maintain the program, since if the value needs to be changed you just change the definition of the constant (which is usually at the start of the program code) rather than having to find and change every place that the value is used.

It also makes the code more readable, since the constant's name is more meaningful than seeing the raw value, which can easily be confused with other values.

Python does *not* have any statement to define a constant. Why not?

It was deemed unnecessary – the usefulness of a constant described above can all be achieved by simply declaring a variable and just not changing its value, e.g.

```
MAX_SIZE = 100
```

Python

The UPPER_CASE naming indicates that it should be treated as a constant (and hence the code should not try to assign a new value to it at any point in the program), however there is nothing that will actually stop you if you try.