Using the Command Line

1. **print(“hello world”)**

|  |
| --- |
| hello world |

1. **10 + 10**

|  |
| --- |
| 20 |

1. **2005.5 / 1.5**

|  |
| --- |
| 1337.0 |

1. **chr(104),chr(101),chr(105)**

|  |
| --- |
| (“h”, “e”, “i”) |

1. **import this**

|  |
| --- |
| The Zen of Python, by Tim Peters  Beautiful is better than ugly.  Explicit is better than implicit.  Simple is better than complex.  Complex is better than complicated.  Flat is better than nested.  Sparse is better than dense.  Readability counts.  Special cases aren't special enough to break the rules.  Although practicality beats purity.  Errors should never pass silently.  Unless explicitly silenced.  In the face of ambiguity, refuse the temptation to guess.  There should be one-- and preferably only one --obvious way to do it.  Although that way may not be obvious at first unless you're Dutch.  Now is better than never.  Although never is often better than \*right\* now.  If the implementation is hard to explain, it's a bad idea.  If the implementation is easy to explain, it may be a good idea.  Namespaces are one honking great idea -- let's do more of those! |

Next, let’s learn how to repeat statements. Type the following line of code and execute it

|  |
| --- |
| **"".join(map(chr,(109, 111, 114, 119, 97, 121)))** |

When we execute this we can see there is an error somewhere. To fix this we need to change 109 to 110. Instead of retyping the entire line simply press the up arrow key on your keyboard ↑, this will retype the previous line you used! Now to change 109 to 110 by moving the command line cursor \_ using the left arrow key ← until it to the right of the 9, use backspace to delete 09 and replace it with 10. Press enter to execute the edited command.

1. What was the resulting output?

|  |
| --- |
| ‘norway’ |

# Command line arguments

After reviewing the chapter on command line arguments in the notes answer the following:

1. What is a command line argument?

|  |
| --- |
| Text written after the script file for giving it information for it’s main functionality |

1. What is a command line option?

|  |
| --- |
| Flags given to the program to change how the script runs and is identified by either one or two dashes. |

1. How can we get help when executing any Python script?

|  |
| --- |
| Running the script followed by –help |

1. Can command line arguments appear in any order we wish? Why?

|  |
| --- |
| Command line arguments should not appear in any order as the script should be able to know how to interpret each section of the arguments, command line options on the other hand are identified by their flags and will be easier for the script to separate. |

1. Give your own example of a bad use of command line arguments.

|  |
| --- |
| python somescript.py from(argument) from(file) to(argument) to(file) |