

School Nova Computer Science

Classwork 2
By Oleg Smirnov



Work organization



- Organize your folders in a way that will make it easy for you to find your code. For example, create a Folder “CS101” (or “CS201”). Inside, create folders “Classwork” and “Homework”. You may also want to create additional folders organized by date.
- Always save your Python script using a file name that accurately describes the code. For example, Smirnov_homework2_version1.py.
- Do *NOT* accept the default filenames such as temp.py, untitled0.py, and so on.
- Save your Python script folder in the corresponding folder (for example, “Homework”), where you can easily find it later. You may need it later, or you may need to upload your file.



Coding, first tips, p1

- We *generally* follow the mainstream conventions as outlined in Python Enhancement Proposal (PEP): <https://www.python.org/dev/peps/pep-0008/>
- Add comments to your code:
 - # This is a comment. Python will ignore anything after the # symbol. (Unless # is placed inside a string!)
- Comments can be very useful if you want to save information for other people, other programmers or your teacher, or even yourself at a later date.
- Maintain consistent style throughout your code. For example, use either ‘single-quoted strings’ or “double-quoted strings” throughout your code. Do not mix. I recommend using double-quoted strings.
- Limit lines to 79 characters. We will explore ways to wrap long lines.

Coding, first tips, p2



- For example, add spaces around math operations:

write “a + b” instead of “a+b”

- Add a space after a comma, colon, semicolon (but note before):

write “a, b = 1, 2” instead of “a,b=1,2”

- Avoid unnecessary spaces (or extra spaces at the end of a line):

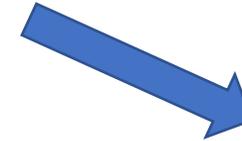
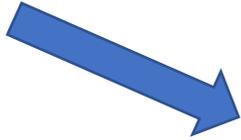
write “month = 9” instead of “month = 9 ”

- Name your variables, functions, classes, and so on in a way that makes intuitive sense. For example,

write: name = “Oleg” instead of n = “Oleg”

To Be Continued

Objects and variables, p1

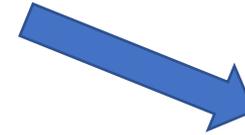
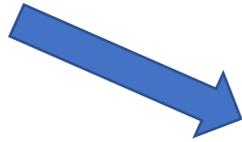


Play Food 	Cooking Toys
Blocks 	Baby Things
Wii Accessories 	Books
Transformers 	Transformers
Bouncy Balls 	Board Games

Copyright © Kelly Turner 2011-2012
www.kellyturner.com

- In Python, when you create a variable, you actually create an **object** and a **reference** to it: for example, $a = 5$, where 5 is an object and a is a reference to it (you can think of a reference as a name).
- This is especially important to understand once we start working with Python lists. We will revisit this discussion at that time.
- If you change the value of a variable, you create a new object!
- In Python you don't need to declare what kind of an object it is. Python assigns a type to the object automatically based on the content.

Objects and variables, p2



Play Food 	Cooking Toys 
Blocks 	Baby Things 
Wii Accessories 	Books 
Transformers 	Transformers 
Bouncy Balls 	Board Games 

Copyright © Kelly Turner 2011-2012
www.kellyturner.com

- `id()` shows the unique integer identification of any object: `id(a)`.
- Try different examples to see that variable names are like stickers or labels.
- Are there objects without labels? Not for long. For example,
 `a = 5` (creates an object with value 5 and reference *a*)
 `a = 6` (creates a new object with value 6, the object above with value 5 becomes “garbage” since there is no reference to it)
- Python removes garbage to free memory (“garbage collection”)
- *Note on variable names*: can only contain numbers, letters, and underscore (“_”); cannot start with a number; case sensitive.

Python Data Types, p1



name = "John Smith"	String
year = 2019	Integer
numPi = 3.14159	Float
numPi = "3.14159"	String
result = True result = False	Boolean
result = "True"	String

type(varname) will show the type of the variable varname

Arithmetic operators



Addition	+	
Subtraction	-	
Multiplication	*	
Division	/	
Floor division	//	17 // 5 (returns 3, removes decimal)
Modulus:	%	17 % 5 (returns 2, remainder)
Exponent:	**	2 ** 3 (returns 8, 2 to the power of 3)