|  |  |
| --- | --- |
| **The Python Interface**  100xp  In the Python script on the right, you can type Python code to solve the exercises. If you hit *Submit Answer*, your python script (script.py) is executed and the output is shown in the IPython Shell. DataCamp checks whether your submission is correct and gives you feedback.  You can hit *Submit Answer* as often as you want. If you're stuck, you can click *Get Hint*, and ultimately *Get Solution*.  You can also use the IPython Shell interactively by simply typing commands and hitting Enter. When you work in the shell directly, your code will not be checked for correctness so it is a great way to experiment.  **Instructions**   * Experiment in the IPython Shell; type 5 / 8, for example. * Add another line of code to the Python script: print(7 + 10). * Hit *Submit Answer* to execute the Python script and receive feedback. | # Example, do not modify!  print(5 / 8)  # Put code below here |

|  |  |
| --- | --- |
| When to use Python? 50xp  Python is a pretty versatile language. For what applications can you use Python? Possible Answers     You want to do some quick calculations.      For your new business, you want to develop a database-driven website.      Your boss asks you to clean and analyze the results of the latest satisfaction survey.      All of the above. |  |

|  |  |
| --- | --- |
| **Any comments?**  100xp  Something that Filip didn't mention in his videos is that you can add **comments** to your Python scripts. Comments are important to make sure that you and others can understand what your code is about.  To add comments to your Python script, you can use the # tag. These comments are not run as Python code, so they will not influence your result. As an example, take the comment on the right, # Just testing division: it is completely ignored during execution.  **Instructions**  Above the print(7 + 10), add the comment # Addition works too. | # Just testing division  print(5 / 8)  # Addition works too |

|  |  |
| --- | --- |
| Python as a calculator 100xp  Python is perfectly suited to do basic calculations. Apart from addition, subtraction, multiplication and division, there is also support for more advanced operations such as:   * Exponentiation: \*\*. This operator raises the number to its left to the power of the number to its right: for example 4\*\*2 will give 16. * Modulo: %. It returns the remainder of the division of the number to the left by the number on its right, for example 18 % 7 equals 4.   The code in the script on the right gives some examples. Instructions Suppose you have $100, which you can invest with a 10% return each year. After one year, it's 100×1.1=110100×1.1=110 dollars, and after two years it's 100×1.1×1.1=121100×1.1×1.1=121. Add code on the right to calculate how much money you end up with after 7 years. | # Addition and subtraction  print(5 + 5)  print(5 - 5)  # Multiplication and division  print(3 \* 5)  print(10 / 2)  # Exponentiation  print(4 \*\* 2)  # Modulo  print(18 % 7)  # How much is your $100 worth after 7 years? |