**Python Course**

**Exercises**

**Variables and basic calculations**

|  |  |
| --- | --- |
| 1.1. | Write code that can print the sentence “It works!” or any other sentence that you can think of. |
| 1.2. | Write a program which produces the following output:   |  | | --- | | This is the first line. This is the second line. This line contains a        tab. | |
| 1.3. | Declare three numerical variables with arbitrary values. What is the average of these three values? |
| 1.4. | Given an exchange rate of 1.2409 (euros to American dollars), how many dollars can you buy for 150 euros? Use the in-built round() function to ensure that the floating point number that results from the calculation is shown as an actual amount, indicating the number of cents. The first parameter of 'round()' is the number to be rounded, and the second number is the number of digits beyond the decimal point. |
| 1.5. | Write code which can calculate the number of seconds in seven days. |
| 1.6. | Write code which can convert a temperature of 37 degrees Celcius into Fahrenheit. Use the following formula: F = 1.8 \* C + 32. |

**Flow Control**

|  |  |
| --- | --- |
| 2.1. | Write code which can give information about the century a given year is in. Given the year 1767, for instance, the application must be able to print the following sentence: “The year 1767 is in the eighteenth century”. Limit your algorithm to years in between 1500 and 2010.  Note that it is possible to combine two different Boolean expressions using the word ‘and’. |
| 2.2. | Create a variable and assign it an arbitrary numeric value. Next, write some code which can print the multiplication table for this number. |
| 2.3. | Using the keyword ‘for’ and the range() function, write code which generate the drawing below. N.B. The longest line in the pattern consists of seven stars.   |  | | --- | | \*  \*\*  \*\*\*  \*\*\*\* \*\*\*\*\*  \*\*\*\*\*\*  \*\*\*\*\*\*\* \*\*\*\*\*\* \*\*\*\*\* \*\*\*\*  \*\*\*  \*\* \* | |
| 2.4. | Create two numeric variables.   |  | | --- | | first = 0  second = 1 |   Use these two varibales to generate a Fibonacci sequence. This sequence is created by letting each new number be the sum of the previous two numbers. If the first numbers are 0 and 1, the sequence should continue as ‘1, 2, 3, 5, 8, 13, 21’. The sequence should stop as soon as a number has reached a value higher than 200. |

**Working with strings**

|  |  |
| --- | --- |
| 3.1. | Create two string variables. The first variable must be assigned the value ‘unique’ and the second variable should be assigned the value ‘biodiversity’. Create a third variable with the value ‘university’, by firstly slicing the first two string variables, and by subsequently concatenating the substrings. |
| 3.2. | Create the following two string variables:   |  | | --- | | first = 'vladimir'  last = 'nabokov' |   Using these two existing variables, create a third variable named ‘fullName’ with the following value: “Nabokov, Vladimir”. Note that the first character of the first name and the last name must be in upper case. |
| 3.3. | Create the following two strings:   |  | | --- | | quote1 = 'The truth is rarely pure and never simple'  quote2 = 'Experience is the name we give to our mistakes' |   Which of these two quotes has the highest number of characters? |
| 3.4. | Create a variable named ‘url’ and assign it following value: ‘https://www.universiteitleiden.nl’.  Try to write code which can extract the top domain level (i.e. the country code) from this url.  Tip: this problem can be solved by creating a string slice with the last two characters only, but top domain levels may of course consists of more than two characters. An alternative approach is to make use of the ‘rindex()’ function, which returns the LAST occurrence of a character. |
| 3.5. | Create a variable named ‘filename’ and assign it the value ‘README.txt’. Next, write some code in Python which can extract the filename without the extension. |

**Functions**

|  |  |  |
| --- | --- | --- |
| 4.1. | In a given university course, the final grade is determined by the grade for an essay (70%) and the grade for a presentation (30%).  Write two functions: (1) one which can calculate the final grade, given a set of partial grades. Grades must be rounded to integers. For example: 5.4 must become 5 and 6.6 must become 7. N.B. This is the precisely the way in which the in-built ‘round()’ function rounds numbers (2) Write a second function which can determine whether a given grade is at a pass level (i.e. higher than 6). This function must return either the text ‘Pass’ or ‘Fall’.  Use the code below as a basis.   |  | | --- | | essay = 7.0  presentation = 8.5  final = calculateMark(essay,presentation)  print( "final grade: {} ({})".format( final , isPass(final) ) )  essay = 4.5  presentation = 5.5  final = calculateMark(essay,presentation)  print( "final grade: {} ({})".format( final , isPass(final) ) ) | |
| 4.2. | Import the math library, as follows:   |  | | --- | | from math import \* |   This command simply imports all the available functions from the math module. Use the functions ‘log10()’, ‘pow()’, ‘sqrt()’ and ‘cos()’ to generate the following numbers:   * The base-10 logarithm of 5. * 3 raised to the power of 4 * The square root of 144 * The cosine of 60 radians.   For more details, consult the documentation of the ‘math’ module at <https://docs.python.org/3/library/math.html> |
| 4.3. | Following Pythagoras’ theorem (A2 + B2 = C2), calculate the length of the hypothenuse in a right trangle in which the other two sides have a length of 6 and 7. Make use of the math module. |

**Lists**

|  |  |  |
| --- | --- | --- |
| 5.1. | The list below contains the titles of first twelve plays written by William Shakespeare.   |  | | --- | | plays = [ 'Comedy of Errors' ,  'Henry VI, Part I' ,  'Henry VI, Part II' ,  'Henry VI, Part III' ,  'Richard III' ,  'Taming of the Shrew' ,  'Titus Andronicus' ,  'Romeo and Juliet' ,  'Two Gentlemen of Verona' ,  'Love\'s Labour\'s Lost' ,  'Richard II' ,  'Midsummer Night\'s Dream' ] |   Add the following two titles to this list:   *Macbeth Othello*  Next, count the number of plays in this list and print the titles of the first and the last plays in the list, using the index of these items.  Print the full list in alphabetical order using the ‘for’ keyword. |
| 5.2. | Convert the following quotation from E.M. Forster’s *A Room with a View* to a list of words, using the split() method. This function can convert a string into a list, based on the spaces that occur in this string. Use this list to give information about the total number of words in this quote, and the number of occurrences of the word “place”.   |  | | --- | | quote = "We cast a shadow on something wherever we stand, and it is no good moving from place to place to save things; because the shadow always follows. Choose a place where you won’t do harm - yes, choose a place where you won’t do very much harm, and stand in it for all you are worth, facing the sunshine." | |
| 5.3. | Copy the code below.   |  | | --- | | sites = ['https://www.universiteitleiden.nl' , 'https://www.stanford.edu', 'https://www.uu.nl' , 'http://www.ox.ac.uk', 'https://www.rug.nl' , 'https://www.hu-berlin.de', 'https://www.uva.nl'] |   Write some additional code which can select the Dutch websites from this list.  Tip: You may want to reuse some of the code developed for exercise 3.4. |
| 5.4. | Create a list named ‘vowels’, using the following code:   |  | | --- | | vowels = ['a', 'o', 'u', 'i', 'e'] |   Write code in Python which can calculate the number of vowels and the number of consonants in the words stored in the following list. In your code, make use of the list of vowels that you created.   |  | | --- | | words = ['acknowledge', 'beautiful', 'contemporary', 'display', 'equivalent'] | |
| 5.5. | Create two lists:   |  | | --- | | first = [4, 9, 1, 17, 11, 26, 28, 54, 63]  second = [9, 9, 74, 21, 45, 11, 63, 28, 26] |   Write code which can extract those numbers from the first list which are NOT in the second list. |
| 5.6. | Create a list named ‘numbers’, using the following code:   |  | | --- | | numbers = [ 4, 7, 26, 38, 43, 67, 82, 94, 111, 126, 137 ] |   Write code in Python which can print the odd numbers only. In your code, make use of the modulo operator (‘%’). |

**Dictionaries**

|  |  |  |
| --- | --- | --- |
| 6.1. | The code below creates a new dictionary. This dictionary connects a number of ISBNs to the titles of the books they identify.   |  | | --- | | isbn = {  9780143105985 : 'White Noise' ,  9780241984536 : 'Libra' ,  9781925480665 : 'Mao II' ,  9781447289395 : 'Underworld' ,  9780743595728 : 'The Body Artist' ,  9781925480665 : 'Cosmopolis' ,  9780330524919 : 'Falling man' ,  9781439169971 : 'Point Omega'  } |  * Add the novel *Zero K* to the dictionary. This novel has ISBN13 9781501138072. * Write some code which can print the title that corresponds to ISBN 9781447289395. * Print a list of all the novels. Display both the ISBN and the title. |
| 6.2. | Using the dictionary ‘data’ below, print the following sentence: “Louis Elsevier was a printer. He was born in 1540 in Leuven and died in 1617 in Leiden.”   |  | | --- | | data = dict()  data["firstName"] = 'louis'  data["lastName"] = 'elsevier'  data["profession"] = 'printer'  data["yob"] = 1540  data["yod"] = 1617  data["pob"] = 'leuven'  data["pod"] = 'leiden' | |
| 6.3 | Copy the code below in your code editor. Using this dictionary, named ‘eu’, print a  sentence which gives information about the current number of countries in  the EU. Add some code which can print a list of all the countries of the EU in  alphabetical order. Finally, print a list of all the capitals of these countries,  also in alphabetical order.   |  | | --- | | eu = {  'Italy':'Rome' , 'Luxembourg':'Luxembourg' , 'Belgium':'Brussels' , 'Denmark':'Copenhagen' , 'Finland':'Helsinki' , 'France':'Paris' , 'Slovakia':'Bratislava' , 'Slovenia':'Ljubljana' , 'Germany':'Berlin' , 'Greece':'Athens' , 'Ireland':'Dublin' , 'Netherlands':'Amsterdam' , 'Portugal':'Lisbon' , 'Spain':'Madrid' , 'Sweden':'Stockholm' , 'Cyprus':'Nicosia' , 'Lithuania':'Vilnius' , 'Czech Republic':'Prague' , 'Estonia':'Tallin' , 'Hungary':'Budapest' , 'Latvia':'Riga' , 'Malta':'Valetta' , 'Austria':'Vienna' , 'Poland':'Warsaw' , 'Croatia':'Zagreb' ,'Romania':'Bucharest' , 'Bulgaria':'Sofia' } | |

**Working with files and folders**

|  |  |  |
| --- | --- | --- |
| 7.1. | Download the following file: <https://edu.nl/ggwtf>  The file contains the text of Sonnet 116 by William Shakespeare. Write code which can generate line numbers. The output should look as follows:   |  | | --- | | 1. Let me not to the marriage of true minds  2. Admit impediments. Love is not love  3. Which alters when it alteration finds,  4. Or bends with the remover to remove:  […] | |
| 7.2. | Using the file containing the text of Sonnet 116 (the download link can be found under exercise 7.1), create a new file named ‘data.txt’, which gives information about the following:   * The total number of characters in the file * The number of lines * The average number of characters per line * The longest line of the sonnet (i.e. the line with the highest number of characters) |
| 7.3. | Using the ‘listdir()’ function from the os library, write code which can list all the files in a given folder on your own computer. |

**Regular expressions**

|  |  |
| --- | --- |
| 8.1. | Download “P.B. Shelley's Complete Poems” from the following address:  <https://edu.nl/r6dn8>  Regular expressions can be used to find verse lines with specific properties. Write a program in Python which can identify verse lines with the following features:   * Lines containing the word "fire". * Lines that containing either the word "sun" or to word "moon". * Use a single regular expression to identify these lines. * All the lines which contain either the singular or the plural form of "star". * All the lines which contain either the singular or the plural form of "leaf". * Lines with words ending in in "ly". * All the lines which contain a question mark. * Lines ending in the character combination "ain". * Cases of alliteration on "br" (or, in other words, all the lines which contain at least two words that begin with "br") |
| 8.2. | Download the file “bibliography.txt” from  <https://edu.nl/t449h>  This file contains a list of articles, formatted according to the APA citation style. For each title, try to extract the year of publication, the title and the name of the journal. |
| 8.3. | Download the file “tweets.txt” from  <https://edu.nl/cvge6>  This file contains a number of tweets containing the hashtag ‘#universiteitleiden’, obtained using the Twitter API. Extract all the usernames and all the hashtags form these tweets, using regular expressions. |

**Data analysis with Pandas**

|  |  |
| --- | --- |
| 11.1. | Download the CSV file ‘nobel.csv’:  <https://edu.nl/3xmbd>  Using the data in this file, try to answer the following questions:   * How many Nobel laureates are described in this file? * How many of these are from the Netherlands? What are the names of the Dutch Nobel prize winners? * How many female Nobel laureates are there in this data set? What are their names? * Who was the youngest person ever to receive the Nobel Prize? Who was the oldest person? Tip: create a new column called ‘Age’, in which the year in which the prize was awarded is subtracted from the year of birth. * Create a list of all the unique countries, and calculate the number of Nobel laureates per country. * For each discipline, calculate the average age of the Nobel laureates. |
| 11.2. | Download the Excel file names ‘Course.xsl’ from the following address:  <https://github.com/peterverhaar/PythonCourse/raw/master/Texts/Grades.xlsx>  This spreadsheet contains the partial grades received in a course. Write code which can calculate and print the final grade for each student. The essay and the exam both count for 40% the grade. The remaining 20% is determined by the presentation.  Additionally, answer the following questions:   * Which student has received the highest mark for the essay? * Which student scored worst on the exam? * How many studens have received a 6 or lower for the essay? |