{

"cells": [

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Python Assignment - 1\n",

"\n",

"### Total - 30 marks"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"## Variables and data types"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 1 a)\_\_ Create four variables and assign values with different data types to them"

]

},

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {},

"outputs": [],

"source": [

"#int, float, String, boolean\n"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 1 b)\_\_ Print the variables using Python's built-in function print(\_String\_)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 1 c)\_\_ Print the data type of each variable using Python's built-in function type()"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 2\_\_ How do we use variables? Explore with arithmetic operators (+, -, \*, /)!"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"8\n"

]

}

],

"source": [

"# Example\n",

"first\_value = 3\n",

"second\_value = 5\n",

"\n",

"result = first\_value + second\_value\n",

"print(result)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"## Example: Lists"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"metadata": {},

"outputs": [],

"source": [

"fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'mango', 'grapes']"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'orange'"

]

},

"execution\_count": 4,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"fruits[0]"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'apple'"

]

},

"execution\_count": 5,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"fruits[1]"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'grapes'"

]

},

"execution\_count": 6,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"fruits[-1]"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"We can also specify a range of indexes specifying where to start and end the range"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"['pear', 'banana', 'kiwi']"

]

},

"execution\_count": 7,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"fruits[2:5]"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"['pear', 'banana', 'kiwi', 'mango', 'grapes']"

]

},

"execution\_count": 8,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"fruits[2:]"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"['orange', 'apple']"

]

},

"execution\_count": 9,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"fruits[:2]"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 3 a)\_\_ Create a list with names of your friends and print the name of the first, second and last person"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 3 b)\_\_ Print your list of names using different ranges"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 4 a)\_\_ Count how many names you have added to the list using len(\_[...]\_) "

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 4 b)\_\_ Add a new name to your list using append(\_item\_)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 4 c)\_\_ Print the list. Does it look as expected?"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 4 d)\_\_ Add a new name on the \_third\_ position in your list using insert(\_position\_, \_item\_)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 4 e)\_\_ Print the list. Does it look as expected?"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 4 f)\_\_ Remove a name from the list using remove(\_item\_)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 4 g)\_\_ Print the list. Does it look as expected?"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Example: For loops"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"orange\n",

"apple\n",

"pear\n",

"banana\n",

"kiwi\n",

"mango\n",

"grapes\n"

]

}

],

"source": [

"for fruit in fruits:\n",

" print(fruit)"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"oranges\n",

"apples\n",

"pears\n",

"bananas\n",

"kiwis\n",

"mangos\n",

"grapess\n"

]

}

],

"source": [

"for fruit in fruits:\n",

" fruit = fruit + 's'\n",

" print(fruit)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 5 a)\_\_ Print each name in your list of names"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 5 b)\_\_ Add a small change to each name in your list"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Example: Dictionaries"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"metadata": {},

"outputs": [],

"source": [

"countries = {\"sweden\" : \"stockholm\", \"denmark\" : \"copenhagen\", \"norway\" : \"oslo\"}"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"stockholm\n"

]

}

],

"source": [

"capital = countries[\"sweden\"]\n",

"print(capital)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 6 a)\_\_ Get the capital of Denmark and Norway using the dictionary \_countries\_"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 6 b)\_\_ Add a new key-value pair to the dictionary \_countries\_"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 6 c)\_\_ Print the dictionary. Does it look as expected?"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 6 d)\_\_ Update Sweden's capital to Malmo"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 6 e)\_\_ Print the dictionary. Does it look as expected?"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 6 f)\_\_ Remove the key-value pair Sweden-Stockholm from the dictionary"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 6 g)\_\_ Print the dictionary. Does it look as expected?"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 6 g)\_\_ Print each key-value pair in the dictionary countries"

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Country: sweden, Capital: stockholm\n",

"Country: denmark, Capital: copenhagen\n",

"Country: norway, Capital: oslo\n"

]

}

],

"source": [

"for key, value in countries.items():\n",

" print(f\"Country: {key}, Capital: {value}\")"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Example: Functions"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"def function\_name(param1, param2):\n",

" # Block of code"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"We will start with creating a function that can add two numbers."

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"def add\_two\_numbers(number1, number2):\n",

" result = number1 + number2\n",

" return result"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"Let's try to use it!"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"add\_two\_numbers(1,2)"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": [

"add\_two\_numbers(5,20)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 7 a)\_\_ Create a function that can calculate the difference between two numbers"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 7 b)\_\_ Use the function. Does it return the expected results?"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 7 c)\_\_ Create a function that can multiply two numbers"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 7 d)\_\_ Use the function. Does it return the expected results?"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 7 e)\_\_ Create a function that can divide two numbers"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"\_\_Exercise 7 f)\_\_ Use the function. Does it return the expected results?"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {},

"outputs": [],

"source": []

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.8.8"

}

},

"nbformat": 4,

"nbformat\_minor": 2

}