untitled5

September 9, 2024

0.1 PYTHON VARIABLES

Varibles are an essential part of python.They allow us to easily store, manipulate, and reference data throughout our projects.

WHAT IS A VARIABLE IN PYTHON?

Varibles are essential for holding onto and referencing values throughout our application. By storing a value into a variable, you can reuse it as many times and in whatever way you like throughout your project.

In python, varibles are created the moment you give or assign a value to them.

HOW DO I ASSIGN A VALUE TO A VARIBLE?

Assighing a value to a variable in python is an easy process. you simply use the equal sign = as an assignment operater, followed by the value you want to assign to the variable. Here's an example...

[13]:
```
fav_place = "EDINBHERG"
why = "INFRASTRUCTURE"
year = 8500
A = "BC"
print("my fav place is", fav_place, "because of its", why, "and it was founded␣
 ↪in", year, A)
```

```
my fav place is EDINBHERG because of its INFRASTRUCTURE and it was founded in
8500 BC
```

In this example, we've created four variables: fav_place, why, year, and A. we've assigned the string value "EDINBHERG" to the fav_place and another string value "INFRASTRUCTURE" to why and an integer value to the varible year and a another string value to the variable A.

- Varibles in python are case-sensitive. in another words, we have to be observent when we are creating the variables, because "fav_place" will be different variable than "FAV_PLACE" even though they include the same letters.
- varibles names that use multiple words in python should be separated with an underscore _ . for example, a variable named "site name" should be written as "sitename"._This convention is called **snake case**.

1 HOW SHOULD I NAME MY VARIABLE:

There are some rules to follow when naming python variables.

some of these are hard rules that must be followed, otherwise your program will not work, while others are known as conventions. This means, they are more like suggestions.

VARIABLE NAMING RULES

- variable names must start with a letter or an underscore_ character.
- variavle names can only contain letters,numbers, and underscores.
- variable names cannot contain spaces or special characters.

[15]:
```
my_age = 21 # valid
course = "DATA SCIENCE" # valid
TIME_PERIOD = "SIX MONTHS" # valid
print(my_age)
print(course)
print(TIME_PERIOD)
```

```
21
DATA SCIENCE
SIX MONTHS
```

VARIABLE NAMING CONVENTIONS

- variable names should be descriptive and not too short or too long.
- use lowercase letters and underscore to separate words in variables names (known as "snake_case").

WHAT DATA TYPES CAN PYTHON VARIABLES HOLD?

One of the best feauters of python is its flexibility when it comes to handling various data types.

Python varibles can hold various data types, including integers, floats, strings, booleans, tuples and lists.

WHAT OPERATIONS CAN BE DONE IN PYTHION USING THE VARIBLES

MATHEMATICAL OPERATIONS

Its possible to perform basic mathematic operations with variables, such as addition, subbtraction, multiplication, and division:

[19]:
```
# Arthematic operations
a = 50
b = 60

sum = a + b
difference = a - b
product = a * b
quotient = a / b
print(sum, difference, product, quotient)
```

```
110 -10 3000 0.8333333333333334
```

1.1 VARIABLE SCOPE

The scope of a variable refers to the part of a program where the variable can be acessed and modified. in python, there are two main types of variables scope;

GLOBAL SCOPE

Variables defined outside of any function or class have a global scope. They can be accessed and modified throughout the program, including within functions and classes.

LOCAL SCOPE

Variables defined within a function or class have a local scope. Thye can only be accessed and modified within that function or class.

VALID VARIABLES

[20]:
```
name = "SHAIK.FAHIM AHMED"
print(name)
```

```
SHAIK.FAHIM AHMED
```

[21]:
```
age = 21
print(age)
```

```
21
```

[22]:
```
fav_hobbie = "singing songs"
print(fav_hobbie)
```

```
singing songs
```

[23]:
```
_m_y_heigh_t = "6.1"
print(_m_y_heigh_t)
```

```
6.1
```

[24]:
```
name1 = "DATA SCIENCE"
print(name1)
```

```
DATA SCIENCE
```

[27]:
```
HOW_are_you = "I am fine"
print(HOW_are_you)
```

```
I am fine
```

[28]:
```
a = 10
b = 25
print(a + b)
print(a < b)
print(a > b)
```

```
print(a * b)
print(a - b)
print(a % b)
print(a == b)
print(a != b)
```

```
35
True
False
250
-15
10
False
True
```

[29]:
```
first_name = "shaik.Fahim"
last_name = "Ahmed"
full_name = first_name + " " + last_name
print(full_name)
```

```
shaik.Fahim Ahmed
```

[31]:
```
#Exponentiation
a = 30
b = 40
power = a ** b
print(power)
```

```
1215766545905692880100000000000000000000000000000000000000000000
1215766545905692880100000000000000000000000000000000000000000000
```

[33]:
```
x = y = z = 100
print(x)
print(y)
print(z)
```

```
100
100
100
```

[35]:
```
a = b = c = (("apple"), ("banana"), ("custard apple"))
print(a)
print(b)
print(c)
```

```
('apple', 'banana', 'custard apple')
('apple', 'banana', 'custard apple')
('apple', 'banana', 'custard apple')
```

INVALID VARIABLES

[36]:
```
3name = "faheem"
print(3name)
```

```
  File "<ipython-input-36-61819d7b0e9a>", line 1
    3name = "faheem"
    ^
SyntaxError: invalid decimal literal
```

[37]:
```
fav place = "gym"
print(fav place)
```

```
  File "<ipython-input-37-c1623b0d3790>", line 1
    fav place = "gym"
        ^
SyntaxError: invalid syntax
```

[38]:
```
@my_id = "faheem@123"
print(@my_id)
```

```
  File "<ipython-input-38-5a2c8fe37687>", line 1
    @my_id = "faheem@123"
     ^
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
```

[40]:
```
if = "I AM LOOSER"
print(if)
```

```
  File "<ipython-input-40-2f53651b3655>", line 1
    if = "I AM LOOSER"
       ^
SyntaxError: invalid syntax
```

[41]:
```
def = "data science"
print(def)
```

```
  File "<ipython-input-41-f238841b4d3e>", line 1
    def = "data science"
        ^
```

```
SyntaxError: invalid syntax
```