basics of python assignment

September 10, 2024

#python_basic_assignment 1

1. Explain the key features of Python that make it a popular choice for programming

Ans:- Python is a dynamic, high-level, free open source, and interpreted programming language.It supports object-oriented programming as well as procedural-oriented programming.In Python, we don't need to declare the type of variable because it is a dynamically typed language. For example, x = 10 Here, x can be anything such as String, int, etc.

The key features of Python are:- 1. Easy to Code 2. Open Source & Free Software 3. Support for GUI 4. Object-Oriented Methodology 5. High-Level Language 6. Highly Portable Language 7. Integrated by nature 8. Extremely Dynamic 9. Extensive Array of Libraries

2. Describe the role of predefined keywords in Python and provide examples of how they are used in a program.

Ans:- Keywords are used to declare variables, create functions, and control the flow of information in a program. They are also used to identify the scope of variables, define classes, and control the flow of execution.

[1]:
```
# if, elif & else
age = 20
if age>=18:
    print("adult")
else:
    print("minor")
```

```
adult
```

[2]:
```
# Loops >> for loop
for i in range(4):
    for j in range(i+1):
        print("*", end = "")
    print()
```

```
*
**
***
****
```

[3]:
```
#  >> while loop
row = 1
while row <= 4:
    col = 1
    while col <= row:
        print("*", end = "")
        col += 1
    print()
    row += 1
```

```
*
**
***
****
```

[4]:
```
# function keyword
def greet(name):
    return f"Hello, {name}"

print(greet("Charu"))
```

```
Hello, Charu
```

3.Compare and contrast mutable and immutable objects in Python with examples.

Ans:-3.In Python, objects are categorized into two main types based on their mutability: mutable and immutable. Here's a comparison of the two, along with examples to illustrate their differences.

Mutable Objects Definition: Mutable objects can be changed after their creation. This means you can modify their content without creating a new object. examples - List, dictionaries, sets

Immutable Objects Definition: Immutable objects cannot be changed once they are created. Any modification results in the creation of a new object.

examples- Tuple, string, Numbers

examples for mutble objects:-

[5]:
```
#list
li = [1, 2, 3]
print(li)

li[0] = 10
print(li)
```

```
[1, 2, 3]
[10, 2, 3]
```

[6]:
```
#dict
my_dict = {'a': 1, 'b': "Charu"}
print(my_dict)
```

```
my_dict['a'] = "pwskills"
print(my_dict)
```

```
{'a': 1, 'b': 'Charu'}
{'a': 'pwskills', 'b': 'Charu'}
```

[7]:
```
my_tuple = (1, 2, "pwskills")
print(my_tuple)


my_tuple=10
```

```
(1, 2, 'pwskills')
```

[8]:
```
# string
a  = "CHARI"
print(a)

# Attempting to modify a string results in a new string
b= a.replace('I', 'U')
print(b)
```

```
CHARI
CHARU
```

[9]:
```
id(a)
```

[9]: 129239728268464

[10]:
```
id(b)
```

[10]: 129239728275376

4.Discuss the different types of operators in Python and provide examples of how they are used.

Ans:-Operators are special symbols used to perform operations on variables and values. Python supports several types of operators, each serving a different purpose.

Exaples:-

[11]:
```
x = 10
y = 9
# addition
print(x + y)
# substraction
print(x - y)
# multiplication
print(x * y)
```

```
# division
print(x / y)
# floor division/
print(x // y)
# modulus
print(x % y)
# Exponentiation
print(x ** y)
```

```
19
1
90
1.1111111111111112
1
1
1000000000
```

Comparison Operators

[12]:
```
x = 4
y = 8
print(x==y)  # equal to
print(x!=y)  # not eual to
print(x>y)  # greater than
print(x<y)  #less tham
print(x<=y) # less than equal to
print(x>=y) #greater than equal to
```

```
False
True
False
True
True
False
```

Assignment Operators

Bitwise Operators

[13]:
```
x = 8
y = 9

# Bitwise AND
print(x & y )

# Bitwise OR
print(x | y)

# Bitwise XOR
```

```
print(x ^ y)


# Left shift
print(x << y)

# Right shift
print(x >> y)
```

```
8
9
1
4096
0
```

5. Explain the concept of type casting in Python with examples.

Ans:-

Type casting, or type conversion, is the process of changing a variable from one data type to another. In Python, this is often needed when you want to make sure your data is in the correct format for certain operations or functions. Python offers several built-in functions to help with type casting.

Examples

[14]:
```
a = 6.14
int(a)
```

[14]:
```
6
```

[15]:
```
str_num = "64"
int(str_num)
```

[15]:
```
64
```

[16]:
```
flag = True
int(flag)
```

[16]:
```
1
```

[17]:
```
num = 6
float(num)
```

[17]:
```
6.0
```

[18]:
```
str_num = "4.14"
float(str_num)
```

[18]:
```
4.14
```

[19]:
```
flag = False
float(flag)
```

[19]:
```
0.0
```

[20]:
```
num = 500
str(num)
```

[20]:
```
'500'
```

[21]:
```
num = 3.14
str(num)
```

[21]:
```
'3.14'
```

6. How do conditional statements work in Python? Illustrate with examples.

Ans:-Ans:-

Conditional statements in Python let you run different blocks of code depending on whether certain conditions are true or false. Python uses if, elif, and else to decide what code to run.

if condition: # Block of code executed if condition is True elif another_condition: # Block of code executed if the above condition is False and this one is True else: # Block of code executed if all previous conditions are False

7. Describe the different types of loops in Python and their use cases with examples.

ans

[1]:
```
l = [1,2,"charu","pwskills"]
for i in l:
    print(i)
else:
    print("This will be executed when while is run successfuly without any␣
 ↪break")
```

```
1
2
charu
pwskills
This will be executed when while is run successfuly without any break
```

[2]:
```
count = 0
while count < 5:
    print(count)
    count += 1
```

```
0
1
2
3
4
```

Use of Break and continue statement

[3]:
```
l = [1,2,"Charu","pwskills"]
for i in l:
        if i == "Nadeem":
            break
        print(i)
```

```
1
2
Charu
pwskills
```

[4]:
```
for i in l:
        if i == "Charu":
            continue
        print(i)
else:
    print("This will be executed when while is run successfuly without any␣
 ↪break")
```

```
1
2
pwskills
This will be executed when while is run successfuly without any break
```

[ ]: