Python
Unit:-1&2 imp
From
Question bank
Solved by :- lakshya bhatt

-:- 1 marker

1. Which of the following is a valid Python variable name?

a) 1_variable

b) variable_name

c) variable-name

d) @variable

Answer: b) variable_name

2. What is the correct syntax to output "Hello World" in Python?

a) echo "Hello World"

b) print("Hello World")

c) printf("Hello World")

d) cout << "Hello World";

Answer: b) print("Hello World")

3. Which of the following is immutable in Python?

a) List

b) Dictionary

c) Tuple

d) Set

Answer: c) Tuple

4. What will be the output of print(type(5/2))?

a) int

b) float

c) double

d) long

Answer: b) float

5. Which keyword is used to define a function in Python?

a) define

b) function

c) def

d) func

Answer: c) def

6. What does len([1,2,3]) return?

a) 1

b) 2

c) 3

d) 4

Answer: c) 3

7. Which operator is used for exponentiation in Python?

a) ^

b) **

c) %

d) //

Answer: b) **

8. What will be the output of bool([])?

a) True

b) False

Answer: b) False

9. What will be the output of print(2 == 2.0)?

a) True

b) False

Answer: a) True

10. What is the output of print("Python"[::-1])?

a) nohtyP

b) Python

c) Pytho

d) Error

Answer: a) nohtyP

-:- question answers

Unit 1: Introduction to Python Programming

1. Explain the features of Python.

Answer:

- **Interpreted Language** – No need for compilation.
- **Easy to Learn & Readable** – Uses simple syntax.
- **Dynamically Typed** – No need to declare variable types.
- **Object-Oriented & Functional** – Supports OOP and functional programming.
- **Extensive Libraries** – Includes NumPy, Pandas, Matplotlib, etc.
- **Platform Independent** – Works on Windows, macOS, and Linux.

---

2. Compare Python with C and Java.

---

3. Which are the built-in data types in Python?

Answer:

- **Numeric**: `int`, `float`, `complex`

- **Sequence**: `list`, `tuple`, `range`
- **Text**: `str`
- **Set**: `set`, `frozenset`
- **Mapping**: `dict`
- **Boolean**: `bool`

---

4. Explain list with an example and explain any 5 methods.

Answer:
A **list** is an ordered, mutable collection in Python.

```
my_list = [1, 2, 3, 4]
print(my_list[1])  # Output: 2
```

Methods:

- `append()` – Adds an element to the end of the list.
- `pop()` – Removes and returns the last element.
- `insert(index, value)` – Inserts a value at a specific index.
- `extend()` – Extends the list with another list.
- `sort()` – Sorts the list in ascending order.

---

5. Differentiate between list and set.

Difference Between List and Set

1. Order - List maintains order, Set is unordered

2. Duplicates - List allows duplicates, Set does not

3. Indexing - List supports indexing, Set does not

4. Mutability - Both List and Set are mutable

5. Performance - Set is faster for searching, List is slower

---

6. Write a Python program to reverse a list.

```
my_list = [1, 2, 3, 4, 5]
my_list.reverse()
print(my_list)  # Output: [5, 4, 3, 2, 1]
```

---

7. What is a dictionary in Python? Explain with an example.

Answer:
A dictionary is a collection of key-value pairs.

```
student = {"name": "John", "age": 21, "course": "Python"}
print(student["name"])  # Output: John
```

---

8. Explain membership operators in Python.

Answer:

- `in` – Returns `True` if a value is in a sequence.
- `not in` – Returns `True` if a value is not in a sequence.
  Example:

```
print("a" in "apple")  # Output: True
print(5 not in [1, 2, 3])  # Output: True
```

---

9. Explain list comprehension with an example.

Answer:
List comprehension is a concise way to create lists.

```
squares = [x**2 for x in range(5)]
print(squares)  # Output: [0, 1, 4, 9, 16]
```

---

10. Write a Python program to find the area of a circle.

```
r = float(input("Enter radius: "))
```

```
area = 3.14 * r * r
print("Area:", area)
```

Unit 2: Conditional & Control Structures

11. Explain the while loop with an example.

Answer:
A **while loop** runs as long as the condition is `True`.

```
i = 1
while i <= 5:
    print(i)
    i += 1
```

Output:

```
1
2
3
4
5
```

---

12. Make a simple calculator using if-else.

```
a = float(input("Enter first number: "))
b = float(input("Enter second number: "))
op = input("Enter operator (+, -, *, /): ")

if op == '+':
    print(a + b)
elif op == '-':
    print(a - b)
elif op == '*':
    print(a * b)
elif op == '/':
    print(a / b)
else:
    print("Invalid operator")
```

---

13. Explain break and continue statements with examples.

Break: Stops the loop immediately.

```
for i in range(5):
    if i == 3:
        break
    print(i)
# Output: 0 1 2
```

Continue: Skips the current iteration and moves to the next.

```
for i in range(5):
    if i == 3:
        continue
    print(i)
# Output: 0 1 2 4
```

---

14. Write a Python program to print the first 10 even numbers using a while loop.

```
i = 2
while i <= 20:
    print(i)
    i += 2
```

---

15. Write a Python program to check if a number is prime or not.

```
num = int(input("Enter a number: "))
if num > 1:
    for i in range(2, num):
        if num % i == 0:
            print("Not a prime number")
            break
    else:
        print("Prime number")
else:
    print("Not a prime number")
```

---

16. Explain the range() function in Python with examples.

Answer:
The `range()` function generates a sequence of numbers.

- `range(5)` → `[0, 1, 2, 3, 4]`
- `range(2, 6)` → `[2, 3, 4, 5]`
- `range(1, 10, 2)` → `[1, 3, 5, 7, 9]`

Example:

```
for i in range(1, 6):
    print(i)
```

Output:

```
1
2
3
4
5
```

---

17. Illustrate the if-elif-else statement with an example.

```
num = int(input("Enter a number: "))

if num > 0:
    print("Positive")
elif num < 0:
    print("Negative")
else:
    print("Zero")
```

---

18. Explain enumerate() function with an example.

Answer:
The `enumerate()` function adds an index to an iterable.

```
fruits = ["apple", "banana", "cherry"]
```

```
for index, fruit in enumerate(fruits):
    print(index, fruit)
```

Output:

```
0 apple
1 banana
2 cherry
```
