python-inbuilt-functions

September 5, 2024

[29]:

```
# 1. print() - Displays output
print("Hello, World!")
```

```
Hello, World!
```

[30]:

```
# 2. len() - Returns the length of an object
print(len("Python"))
```

```
6
```

[31]:

```
# 3. type() - Returns the data type of an object
print(type(5))
```

```
<class 'int'>
```

[32]:

```
# 4. int() - Converts a value to an integer
print(int(4.7))
```

```
4
```

[33]:

```
# 5. float() - Converts a value to a floating-point number
print(float(10))
```

```
10.0
```

[34]:

```
# 6. str() - Converts a value to a string
print(str(100))
```

```
100
```

[35]:

```
# 8. sum() - Returns the sum of an iterable
numbers = [1, 2, 3, 4]
print(sum(numbers))
```

```
10
```

[36]:

```
# 9. min() - Returns the smallest item in an iterable
print(min(3, 1, 2))
```

```
# 10. max() - Returns the largest item in an iterable
print(max(3, 1, 2))
```

```
1
3
```

[37]:

```
# 11. abs() - Returns the absolute value of a number
print(abs(-10))
```

```
10
```

[38]:

```
# 12. round() - Rounds a number to a given precision
print(round(4.567, 2))
```

```
4.57
```

[39]:

```
# 13. pow() - Returns the value of a number raised to the power of another
print(pow(2, 3))
```

```
8
```

[40]:

```
# 14. range() - Generates a sequence of numbers
print(list(range(5)))
```

```
[0, 1, 2, 3, 4]
```

[41]:

```
# 15. sorted() - Returns a sorted list of the specified iterable
print(sorted([3, 1, 2]))
```

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

[42]:

```
# 16. reversed() - Returns a reversed iterator
print(list(reversed([3, 1, 2])))
```

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

[43]:

```
# 17. enumerate() - Returns an enumerate object
names = ['Alice', 'Bob', 'Charlie']
for i, name in enumerate(names):
    print(i, name)
```

```
0 Alice
1 Bob
2 Charlie
```

[44]:

```
# 18. zip() - Combines two or more iterables into a single iterable of tuples
a = [1, 2, 3]
b = ['a', 'b', 'c']
```

```
print(list(zip(a, b)))
```

```
[(1, 'a'), (2, 'b'), (3, 'c')]
```

[45]:

```
# 19. map() - Applies a function to all items in an iterable
def square(x):
    return x * x
print(list(map(square, [1, 2, 3])))
```

```
[1, 4, 9]
```

[46]:

```
# 20. filter() - Filters items out of an iterable based on a function
def is_even(x):
    return x % 2 == 0
print(list(filter(is_even, [1, 2, 3, 4])))
```

```
[2, 4]
```

[47]:

```
# 21. all() - Returns True if all elements in an iterable are true
print(all([True, True, False]))

# 22. any() - Returns True if any element in an iterable is true
print(any([False, False, True]))
```

```
False
True
```

[48]:

```
# 23. dir() - Returns a list of valid attributes of an object
print(dir([]))
```

```
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__',
'__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__getitem__', '__getstate__', '__gt__', '__hash__',
'__iadd__', '__imul__', '__init__', '__init_subclass__', '__iter__', '__le__',
'__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__',
'__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__',
'__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear',
'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse',
'sort']
```

[49]:

```
# 24. eval() - Evaluates a string as a Python expression
x = '1 + 1'
print(eval(x))
```

```
2
```

[50]:

```
# 25. bool() - Converts a value to a boolean
print(bool(0))
```

```
False
```

[51]:

```
# 26. bin() - Converts an integer to a binary string
print(bin(10))
```

```
0b1010
```

[52]:

```
# 27. ord() - Returns the Unicode code for a character
print(ord('A'))
```

```
65
```

[53]:

```
# 28. chr() - Returns the string representing a character for the given Unicode␣
  ↪code
print(chr(65))
```

```
A
```

[54]:

```
# 29. isinstance() - Checks if an object is an instance or subclass of a class
x = 5
print(isinstance(x, int))
```

```
True
```

[61]:

```
# 30. slice() - Returns a slice object
s = slice(1, 4, 6)
print(list(range(10)[s]))
```

```
[1]
```

[56]:

```
# 31. format() - Formats a value with a specified format
print("{:.2f}".format(3.14159))
```

```
3.14
```

[57]:

```
# 32. divmod() - Returns a tuple of quotient and remainder
print(divmod(10, 3))
```

```
(3, 1)
```

[58]:

```
# 33. complex() - Creates a complex number
print(complex(2, 3))
```

```
(2+3j)
```

[59]:

```
# 34. reversed() - Returns a reversed iterator
print(list(reversed([1, 2, 3])))
```

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

[ ]:

```
```