Code Gemini

+91 9373610956
www.codegemini.in
codegeminiedtech@gmail.com

- **How would you convert one data type to another in Python? For example, an integer to a string.**

To convert one data type to another in Python, you use something called "type casting." For example, if you have an integer `5` and want to convert it into a string, you would use the `str()` function like this:

```
num = 5
num_as_string = str(num)  # Converts 5 to "5"
```

Similarly, there are functions like `int()`, `float()`, and `list()` for converting to other types.

- **What is the difference between == and = in Python?**

In Python, `=` is an *assignment operator* that is used to assign a value to a variable. For example:

```
a = 5  # Assigns 5 to a
```

On the other hand, `==` is a *comparison operator* used to check if two values are equal. For example:

```
if a == 5:  # Checks if a is equal to 5
    print("Equal!")
```

- **How does the modulus operator (`%`) work in Python?**

The modulus operator (`%`) returns the *remainder* of a division. For example:

```
10 % 3  # This will give 1 as the remainder
```

It's useful when you want to check if a number is even or odd, among other things. For example, `if x % 2 == 0` checks if a number is even.

- **What is type casting, and why is it important in Python?**

Type casting is the process of converting one data type to another, such as converting an integer to a float or a string. It's important because in Python, different operations require specific types. For example, adding a number to a string directly would give an error, so type casting is necessary to handle such cases.

- **What will be the output of `3 * "Hello"` in Python? Explain why.**

The output will be:

```
HelloHelloHello
```

In Python, multiplying a string by an integer repeats the string that many times. So, `"Hello" * 3` produces "Hello" three times in a row.

+91 9373610956
www.codegemini.in
codegeminiedtech@gmail.com

- **What is string concatenation, and how would you do it in Python?**

String concatenation means *joining two or more strings together*. In Python, you do this using the + operator. For example:

```
first_name = "John"
last_name = "Doe"
full_name = first_name + " " + last_name  # Adds a space between first and
last name
print(full_name)  # Output: John Doe
```