Summarizing key concepts related to Python programming, including variables, constants, inputs, outputs, assignments, arithmetic and comparison operators, data types, casting, and examples.

Python Programming Concepts Documentation

1. Variables

Definition: Variables are labeled containers in programming that store data values.
Usage: They hold information for later use, allowing you to modify and reference data throughout your code.

Example:

```
x = 10

name = "Alice"
```

---

2. Constants

Definition: Constants represent values that should not change throughout the program.
Usage: They are typically written in uppercase letters to indicate that their value is fixed.

Example:

```
PI = 3.14159

MAX_USERS = 100
```

---

3. Data Types

Python supports several data types, including:

- **int**: Integer (whole numbers).
- **float**: Decimal numbers.
- **str**: Text (strings).
- **bool**: Boolean (True/False).
- **list**: A collection of items.
- **dict**: Key-value pairs (dictionaries).

Example:

```
age = 21       # int

price = 10.5   # float
```

```
name = "Alice"  # str
is_student = True # bool
fruits = ["apple", "banana", "cherry"]  # list
person = {"name": "Alice", "age": 21}  # dict
```

---

4. Casting

Definition: Casting is the process of converting one data type to another.
Usage: It allows you to manipulate data types to suit your needs.

Example:

```
num_str = "25"
num_int = int(num_str)  # converts string "25" to integer 25
height = 170.5
height_str = str(height)  # converts float 170.5 to string "170.5"
```

---

5. Inputs

Definition: Inputs are values provided by the user during program execution.
Usage: Use the input() function to collect user input, which is always a string.

Example:

```
user_name = input("Enter your name: ")
age = int(input("Enter your age: "))  # converting input to integer
```

---

6. Outputs

Definition: Outputs display information to the user.
Usage: Use the print() function to show variables or messages.

Example:

```
print(f"Hello, {user_name}! You are {age} years old.")
```

---

7. Assignments

Definition: Assignments are used to give values to variables.
Usage: You use the equal sign = to assign values.

Example:

```
x = 5

y = 10
```

---

8. Arithmetic Operators

Definition: Arithmetic operators perform mathematical operations on numerical values.

Usage: Use these operators to calculate sums, differences, products, etc.

- +: Addition
- -: Subtraction
- *: Multiplication
- /: Division
- %: Modulus (remainder)
- **: Exponentiation (power)
- //: Floor division (division that rounds down)

Example:

```
a = 10

b = 3

sum_result = a + b  # 13
```

---

9. Comparison Operators

Definition: Comparison operators compare values and return True or False.

Usage: Use these operators to evaluate conditions.

- ==: Equal to
- !=: Not equal to
- >: Greater than
- <: Less than
- >=: Greater than or equal to
- <=: Less than or equal to

Example:

```
x = 10

y = 5

print(x > y)  # True
```

---

10. Errors in print() Statements

Here are two examples of print() statements that will lead to errors:

1. **Using an undefined variable**:

```
print(name)  # Error: NameError: name 'name' is not defined
```
