Python I:
Основы

Переменные - это места в памяти, которые выделены для хранения значений. В Python переменные могут хранить различные типы данных, такие как числа, строки, списки, кортежи и т.д.

Оператор присваивания используется для присвоения значения переменной. Он имеет вид =, например:

```
x = 10
name = "John"
```

Типы данных:

В Python есть несколько встроенных типов данных, которые вы можете использовать:

- int - целое число, например, 5
- float - дробное число, например, 3.14
- str - строка, например, 'Hello, world!'
- bool - логический тип данных, который может быть True или False
- list - список, который может содержать элементы разных типов, например, [1, 'apple', True]
- tuple - кортеж, который похож на список, но не может быть изменен после создания, например, (1, 'apple', True)
- dict - словарь, который позволяет хранить значения в формате ключ-значение, например, {'name': 'John', 'age': 30}

В Python вы можете преобразовывать один тип данных в другой. Например, вы можете преобразовать строку в целое число или дробное число. Для преобразования типов данных в Python используются встроенные функции:

- int() - преобразование в целое число
- float() - преобразование в дробное число
- str() - преобразование в строку
- bool() - преобразование в логический тип данных

```
x = 5
y = str(x) # преобразование переменной x в строку
z = float(x) # преобразование переменной x в дробное число
```

Важно помнить, что некоторые типы данных не могут быть преобразованы друг в друга без потери данных. Например, преобразование строки 'hello' в целое число вызовет ошибку, поскольку строка не может быть представлена в виде числа без потери информации.

```
x = 'hello'
y = int(x) # Ошибка: ValueError: invalid literal for int()
with base 10: 'hello'
```

Поэтому, перед преобразованием типов, необходимо убедиться, что данные могут быть корректно преобразованы.

print(), input(), type(), id():

Функция print() в Python используется для вывода данных на экран. Например, чтобы вывести на экран строку 'Hello, world!', вы можете использовать следующий код:

```
print('Hello, world!')
```

Функция input() используется для чтения данных с клавиатуры. Например, чтобы прочитать строку, которую вводит пользователь, вы можете использовать следующий код:

```
name = input('Введите ваше имя: ')
print('Привет, ' + name)
```

Функция type() используется для определения типа переменной, например:

```
 x = 10
 print(type(x))  # <class 'int'>

 name = "John"
 print(type(name))  # <class 'str'>
```

Функция id() используется для определения уникального идентификатора объекта в памяти. Например:

```
x = 10
print(id(x))  # 140704100674832
name = "John"
print(id(name))  # 1890381259040
```

min(), max(), sum():

sum() - вычисляет сумму элементов в списке (или другой итерируемой последовательности):

```
lst = [1, 2, 3, 4, 5]
print(sum(lst)) # 15
```

min() - находит наименьший элемент в списке (или другой итерируемой последовательности):

```
lst = [1, 2, 3, 4, 5]
print(min(lst)) # 1
```

max() - находит наибольший элемент в списке (или другой итерируемой последовательности):

```
lst = [1, 2, 3, 4, 5]
print(max(lst)) # 5
```

Множественное присваивание:

Множественное присваивание - это способ присвоения значений нескольким переменным за одну операцию. Это делается путем размещения переменных слева от знака равенства и значений справа, разделенных запятыми.

```
x, y = 10, 20
```

В этом примере переменной x присваивается значение 10, а переменной y присваивается значение 20.

Это может быть полезно, когда нужно быстро обменять значениями двух переменных без использования третьей временной переменной:

```
x, y = y, x
```

В этом примере значения переменных x и y меняются местами.

Можно использовать и более сложные значения справа от знака равенства, например, кортежи, списки или словари:

```
x, y, z = (1, 2), [3, 4], {'a': 5, 'b': 6}
```

В этом примере переменной x присваивается кортеж (1, 2), переменной y присваивается список [3, 4], а переменной z - словарь {'a': 5, 'b': 6}.

Еще один пример:

```
a, b, c = range(3)
```

В этом примере переменной a присваивается значение 0, переменной b - значение 1, и переменной c - значение 2, которые генерируются функцией range().