Python-0

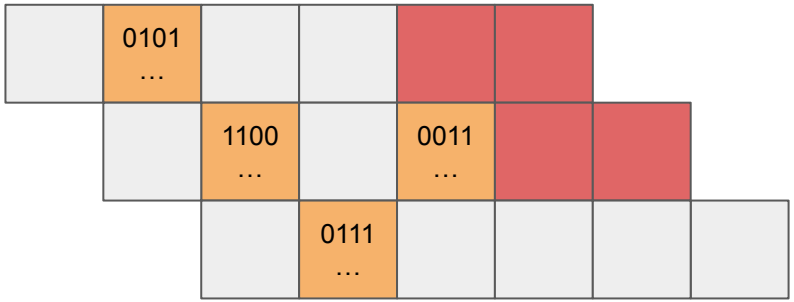
Базовые понятия.


php
JS


Программа
Компилятор
Двоичный код

Компилятор - это как переводчик между языком программирования и языком, который понимает компьютер. Когда программист написал свой код на языке программирования, компилятор переводит его в язык, который компьютер может исполнить. Это как перевести волшебные слова в язык, который понимает твой робот.

Двоичный код - это код состоящий из нулей (0) и единиц (1).



Система счисления (СС) - это способ представления чисел. Мы обычно используем десятичную СС, потому что у нас есть десять цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8 и 9.

В этой системе, например, число "42" означает "4 десятка и 2 единицы".

42 = 4 * 10 + 2

Тем не менее, компьютеры используют двоичную СС, потому что они работают с двумя цифрами: **0** и **1**. Например, в двоичной системе число "42" было бы представлено как "101010". И почему же это так? Как соответственно это произошло?

Есть специальный алгоритм перевода чисел из **X10** -> **X2 СС**.

нужно просто делить число 42 на 2. (*делим исходное число на основание искомого числа и записываем остаток до тех пор, пока неполное частное не будет равно нулю. Полученные остатки записываем в обратном порядке.*)

42 / 2 = 21 (остаток 0)
21 / 2 = 10 (остаток 1)
10 / 2 = 5 (остаток 0)
5 / 2 = 2 (остаток 1)
2 / 2 = 1 (остаток 0)

результат перевода: **101010**

Системы счисления			
Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Пример на закрепление СС:

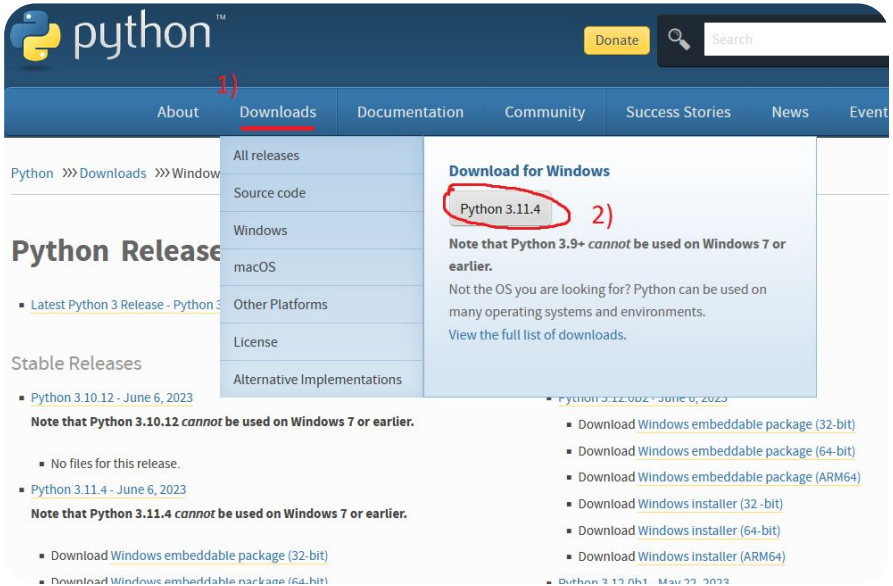
1. Перевести число десятичной системы счисления 421 в двоичную систему счисления.

Деление	Целое частное	Остаток
421 / 2	210	**1**
210 / 2	105	**0**
105 / 2	52	**1**
52 / 2	26	**0**
26 / 2	13	**0**
13 / 2	6	**1**
6 / 2	3	**0**
3 / 2	1	**1**
1 / 2	0	**1**

$421_{10} = 110100101_{2}$

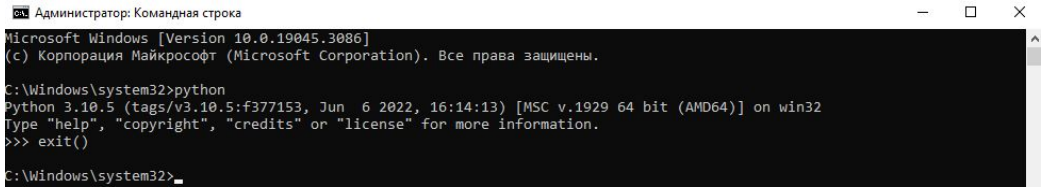
Установка Python. Установка IDE VScode.

Переходим на офиц.сайт Python: https://www.python.org/downloads/ и выбираем здесь раздел (Downloads/Windows или Mac в зависимости от вашей ОС).



Установка Python. Установка IDE VScode.

Установка и выбор параметров:

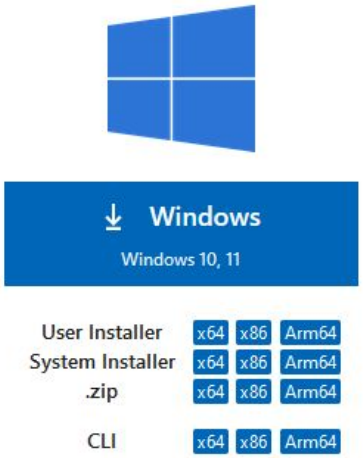


Что такое среда разработки или же IDE?

Среда разработки (IDE) - “ Integrated Development Environment” это рабочее пространство для программиста. IDE предоставляет удобный интерфейс, в котором можно писать код, проверять его на ошибки, выполнять и отлаживать программы, а также управлять проектами. (в том числе и взаимодействие со сторонними сервисами такими как GIT, GITHUB…)

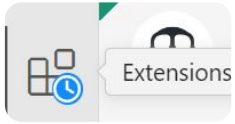
Установка IDE VScode.

https://code.visualstudio.com/download



Открытие IDE VScode.

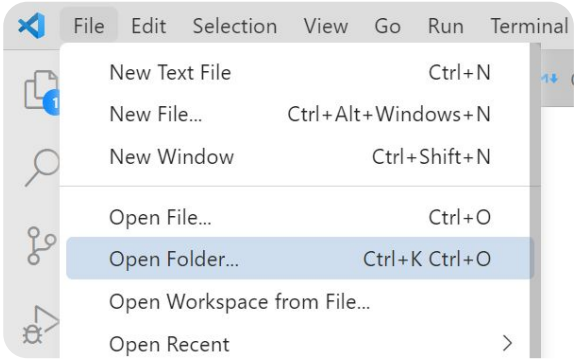
1. На левой крайне панели выбираем раздел Extensions.



2. Пишем в поиске Python.

3. Выбираем первый в списке и нажимаем кнопку Install (Установить)

Создание первого проекта.



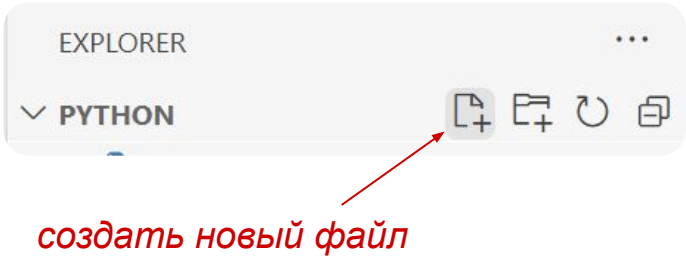
1. На самой верхней панели нажмите на File (Файл) -> Open folder (Открытие папки проекта)

2. Предварительно создайте папку где-то на компьютере (потом только уже ее открывайте)

3. После открытия папки нужно создать файл формата .py (т.е Python файл - файл в котором мы будем уже писать наш код)

Для этого нужно перейти в открытую папку по значку на левой панели.

4. Создать новый файл с именем **main.py**



Запуск кода.

1. Для запуска кода достаточно нажать **CTRL + F5.**
2. А чтобы сохранить изменения в файле **CTRL + S**

Также полезными сочетаниями клавиш являются:

1. **CTRL + C** - копирование чего-либо в буфер обмена
2. **CTRL + V** - вставка из буфера обмена (скопированного или вырезанного кода - текста)
3. **CTRL + X**- вырезать что-то и сохранить в буфер обмена
4. **CTRL + Z** - команда назад (на этап назад, возвращение назад)
5. **CTRL + Y** - команда вперед (на этап вперед)

Принцип программирования.

Простая Яичница (ингредиенты):

- масло (растительное или сливочное);
- 1 яйцо;
- соль.

Шаги приготовления:

- налить или положить масло на сковородку
- разбить яйцо
- посолить
- жарить пока не приготовится

Делим все на отдельные этапы (т.е сложная задача делится на несколько простых подзадач)

Основы Python

Программа на языке программирования, таком как Python, состоит из инструкций, которые компьютер выполняет последовательно.

Пример простой программы Hello world:

```
print("Hello, World!")
```

- print() - это функция, которая выводит текст или значение на экран.
- "Hello, World!" - это строка символов, заключенных в кавычки. В Python строки можно создавать с использованием одинарных (') или двойных (") кавычек.

Хранилище данных. Введение в переменные.

- ведро может быть пустым - None
- ведро может принимать какие-либо предметы (камень, часы…)

т.е ведро здесь в роли переменной которая принимает различные значение (камень, часы и др)

Хранилище данных. Введение в переменные.

Главные параметры которые должна иметь переменная при ее создании:

- *имя переменной*
- *значение переменной*

Хранилище данных. Именование переменных.

Основные правила именования переменных:

1. *Имя переменной может содержать только латинские буквы, числа и символ нижнего подчеркивания*
2. *Имя переменной не должно содержать пробелов*
3. *Имя переменной не должно начинаться с цифры*
4. *Регистр важен: **var** и **Var** это разные переменные*
5. *В Python можно задавать переменные на русском языке (но так делать не рекомендуется - лучше задавать на английском языке)*

Переменные.

Переменные используются для хранения данных (*значений*). Каждая переменная имеет определенный тип данных (*тип значения, например: целое число (int), вещественное число (float), строка (str), булевый (логический True или False) тип (bool) и др*).

Пример создания переменных:

```
age = 18

height = 1.75

string = “Hello, world!”

is_student = True

none_var = None
```

- age, height, string, is_student - это переменные.
- 18 - целочисленный тип данных (*int*).
- 1.75 - тип данных с плавающей точкой (*float*).
- True - логический тип данных (*bool*).
- “Hello, world!” - тип данных строка (*str*).
- None - пустой тип. (исп-ся для создания пустых переменных)

Функция type().

Чтобы узнать тип данных переменной необходимо воспользоваться специальной функцией type().

Пример:

```
variable = 42
print(type(variable)) # <class 'int'>
```

```
float_variable = 3.14
print(type(float_variable))    # <class 'float'>
```

Это может быть полезно, например, при отладке или при написании кода, который должен различать разные типы данных.

Преобразование типов данных.

Преобразование типов данных — это изменение формы данных из одного типа в другой. (*изменить один тип данных на другой, например поменять int на float, или string на int*)

Функции для преобразований:

Функция **int()** используется для преобразования значения в целочисленный тип данных. *Например*, **int**(3.14) вернет значение 3, а **int**("5") вернет значение 5.

Функция **float()** используется для преобразования значения в числовой тип с плавающей точкой. *Например*, **float**(5) вернет значение 5.0, а **float**("2.7") вернет значение 2.7.

Функция **str()** используется для преобразования значения в строковый тип данных. *Например*, **str**(42) вернет строку "42", а **str**(3.14) вернет строку "3.14".

Функция **bool()** используется для преобразования значения в логический тип данных. Любое ненулевое числовое значение или непустая строка будет преобразована в True, а ноль или пустая строка будут преобразованы в False.

Преобразование типов данных. Пример.

Преобразование строки в число

```
строка = "123"

число = int(строка)
```

int(строка) — преобразует строку "123" в целое число 123.

Преобразование числа в строку

```
число = 456

строка = str(число)
```

str(число) — преобразует число 456 в строку "456".

Основные функции.

- input() - функция для ввода данных с клавиатуры.
- print() - функция для вывода данных на экран.

Пример

```
name = input("Введите ваше имя: ")

print("Привет, " + name + "!") # Привет, Алексей!

print("Привет, ", name) # Привет,  Алексей
```

- name - это переменная.

Примеры с print().

Простой вывод сообщение (текст или строка):

```
print("This is my test text!”)
```

Простой вывод значения переменной (без доп текста):

```
a = 5
print(a)
```

Простой вывод значения переменной уже с текстом для красоты:

```
print("value is equal”, a)
```

или так

```
print("value is equal “ + str(a))
```

Пример с переносом строки (\n)

```
print("This is my test text!\n”, “Hello world\n”, “Read file\n”)
```

Примеры с input().

Ввод целого числа с клавиатуры в переменную x.

```
x = int(input("Введите число: "))
```

Ввод вещественного числа (с плав.точкой).

```
float_x = float(input("Введите вещественное число: "))
```

Ввод строки (уже можно преобразование типа данных не делать!).

```
text = input("Введите текст: ")
```

Функция INPUT всегда возвращает *строковый тип данных (str)* именно поэтому мы и не делаем преобразование типа данных когда хотим ввести просто **строку** или **текст**.