

Python

Lesson 1

Numbers, text, print(), and common errors

first_code.py

```
print("Hello, Python!")  
print(2 + 3 * 4)  
print("Alice" * 3)
```

result

```
Hello, Python!  
14  
AliceAliceAlice
```

By the end, you can explain:

operators + - * / //
 **

print sep= and end=

types str int float

errors TypeError /
 SyntaxError

Python can be your calculator

Operators tell Python what calculation to do. Uzbek hint: amal belgisi.

+

addition

qo'shish

$2 + 3 \rightarrow 5$

-

subtraction

ayirish

$10 - 4 \rightarrow 6$

*

multiplication

ko'paytirish

$6 * 7 \rightarrow 42$

/

division

bo'lish

$8 / 2 \rightarrow 4.0$

//

floor division

butun bo'lish

$9 // 2 \rightarrow 4$

**

power

daraja

$2 ** 5 \rightarrow 32$

try_it.py

```
print(2 + 3 * 4)      # 14
print((2 + 3) * 4)    # 20
print(9 / 2)          # 4.5
print(9 // 2)         # 4
```

Order matters

- * and / happen before + and -
- Parentheses () change the order
- / gives a floating-point result
- // removes the decimal part

print() shows the result

print() is a built-in function. It sends text or values to the screen.

default

```
print("Ali")  
print("Vali")
```

output

```
Ali  
Vali
```

sep= (between)

```
print("Ali", "Vali", sep=" +  
")
```

output

```
Ali + Vali
```

end= (after)

```
print("Ali", end="!")  
print("Salom")
```

output

```
Ali!Salom
```

sep = what goes between values • end = what goes after the print

kichik tarjima: between = orasida, after = oxirida

Three types you will use today

A type tells Python what kind of value it is working with.

String

str

text / matn

"Alice"
"42"

Integer

int

whole number / butun son

42
-7

Floating-point

float

decimal number / kasr son

3.14
0.5

type_check.py

```
print(type("Alice")) # <class 'str'>
print(type(42))      # <class 'int'>
print(type(3.14))    # <class 'float'>
```

Text ♦ number? Convert first.

Python does not guess: a string and an integer are different types.

wrong.py

```
name = "Alice"  
age = 42  
print(name + age)
```



right.py

```
print(name + str(age))  
# or  
print(name, age)
```

error

```
TypeError:  
can only concatenate str  
(not "int") to str
```

output

```
Alice42  
Alice 42
```

Casting = changing one type into another

`str(42) → "42"` `int("42") → 42` `float("3.14") → 3.14`

SyntaxError: the quote is not closed

Syntax means the grammar of code. One missing symbol can break the program.

wrong.py

```
print('Alice)
```

right.py

```
print('Alice')  
print("Alice")
```



error

```
SyntaxError:  
EOL while scanning string literal
```

output

```
Alice  
Alice
```

EOL = End Of Line: the line ended before Python found the closing quote.

Rule: every opened quote must be closed – "... " or '... '

Strings can use $\diamond$ and $*$ too

$+$ joins strings. $*$ repeats a string.

concatenation with $+$

```
print("Alice" + "Bob")  
print("Alice" + " " + "Bob")
```

output

```
AliceBob  
Alice Bob
```

multiplication with $*$

```
print("Alice" * 5)  
print("ha" * 3 + "!")
```

output

```
AliceAliceAliceAliceAlice  
hahaha!
```

But string $\diamond$ int does not work: `'Alice'  $\diamond$  42` $\square$

Mini challenge

First predict the output, then run it in Python.

1 `print(5 + 2 * 3)`

What is the result?

2 `print("A", "B", sep="-")`

What does sep do?

3 `print("ha" * 4)`

String multiplication

4 `print("Alice" + str(42))`

Why do we need str()?

5 `print(10 // 3)`

How does // work?

Next lesson: variables, input(), and if/else