Python

Lesson 2

Variables, valid names, multiple assignment, input(), and len()

lesson_2.py

```
name = "Ali"
age = 12
print(name, age)
```

result

```
Ali 12
```

By the end, you can explain:

- **assign** `name = "Ali"`
- **names** `valid or invalid`
- **multiple** `x, y = 1, 2`
- **input ✦ len** `read and count`

Assignment stores a value

The name is on the left. Python evaluates the right side first.

=
assignment
qiymat berish
`score = 10`

←
read it
right to left
`score ← 10`

+
update
yangi qiymat
`score = score + 1`

#
describe
aniq nom
`student_age = 12`

=
overwrite
almashtirish
`score = 99`

?
predict
oldin o'yla
`print(score) → 99`

try_it.py

```
score = 10
score = score + 5
score = 3
print(score)        # 3
```

Follow the value

- First: score gets 10
- Then: score becomes 15
- Then: score is replaced by 3
- print() uses the latest value

Use multiple assignment in one line

Names on the left, values on the right - in the same order.

three variables

```
x, y, z = 1, 2, 3
```

values

```
1  2  3
```

same value

```
a = b = c = 0
```

values

```
0  0  0
```

swap values

```
x, y = 1, 2
x, y = y, x
```

values

```
2  1
```

Left side count = right side count • order matters

Quick check: x, y = 5, 9 → x is 5 and y is 9

Python variable names follow three rules

One word. Letters, numbers, and _. Never start with a number. score ≠ Score.

One word

rule 1

no spaces

```
student_name
firstName
```

Safe characters

rule 2

letters, numbers, _

```
age2
_score
```

Never start 0-9

rule 3

start letter or _

```
student2
2student   ✗
```

name_check.py

```
valid: student_name, age2, _score    invalid: 2age, first name, my-score
```

input() always returns text

Even when the student types digits, Python stores the answer as a string.

wrong.py

```
age = input("Age? ")
next_age = age + 1
print(next_age)
```

right.py

```
age = int(input("Age? "))
next_age = age + 1
print(next_age)
```

error

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

output

```
Age? 12
13
```

Convert only when you need a number

`int(input(...))` → whole number `input(...)` → text

ValueError: the counts do not match

Multiple assignment needs exactly one value for each variable.

wrong.py

```
x, y = 1, 2, 3
```

right.py

```
x, y, z = 1, 2, 3
print(x, y, z)
```

error

```
ValueError:
too many values to unpack
```

output

```
1  2  3
```

Names = Values: keep both sides balanced.

`x, y = 5, 9` ✓ `x, y = 5` ✗ `x, y = 5, 9, 2` ✗

len() counts every character

Letters, spaces, and punctuation are all included.

count a name

```
name = input("Name? ")
print(len(name))
```

output

```
Name? Alice
5
```

spaces count too

```
message = "Hi Ali!"
print(len(message))
```

output

```
7
```

len() returns a number: len('Dombit') → 6

Mini challenge

Predict first. Then run the code and explain what changed.

1	`score = 10; score = score + 2; print(score)`	What prints?
2	`x, y, z = 1, 2, 3`	What value is y?
3	`first name = "Ali"`	Why is this invalid?
4	`name = input("Name? ")`	What type is name?
5	`print(len("Hi Ali!"))`	What prints?

Homework: build a student profile with input(), variables, and len()