مبانی کامپیوتر و برنامه سازی

جلسه ۴

علیرضا اخوان پور
سه‌شنبه - ۴ آبان ۱۳۹۵

1

جنبه‌های زبان:
ساختار دستور (دستور زبان)

- **syntax**
  - **English:** "cat dog boy“ → not syntactically valid
    "cat hugs boy" → syntactically valid
  - **programming language:** "hi"5→ not syntactically valid
    3.2*5 → syntactically valid

2

جنبه‌های زبان:
معنایی ایستا

- **static semantics** is which syntactically valid strings have meaning
  - **English:** "`I are hungry`" → syntactically valid but **static semantic error**
  - **programming language:** `3.2*5` → syntactically valid
    `3+"hi"` → static semantic error

3

جنبه‌های زبان:
معنایی

- **Semantics** is the meaning associated with a syntactically correct string of symbols with no static semantic errors
- **English:** can have many meanings –
  - "Flying planes can be dangerous"
- **فارسی**
  - علی به دوستش گفت که مقاله‌اش منتشر شده است.
  - عفو لازم نیست اعدامش کنید.
- **programming languages:** **have only one meaning but** may **not be what programmer intended**

4

- **syntactic errors**
  - common and easily caught
- **static semantic errors**
  - some languages check for these before running program
  - can cause unpredictable behavior
- no semantic errors but **different meaning than what programmer intended**
  - program crashes, stops running
  - program runs forever
  - program gives an answer but different than expected

5

یک برنامه پایتون

- a **program** is a sequence of definitions and commands
  - definitions **evaluated**
  - commands **executed** by Python interpreter in a shell
- **commands** (statements) instruct interpreter to do something
- can be typed directly in a **shell** or stored in a **file** that is read into the shell and evaluated

6

Object (شئ)

- در پایتون برنامه‌ها اشیاء داده ای (**data objects**) را تغییر میدهند.
- اشیاء دارای نوع (**type**) هستند. این نوع تعیین کننده ی کارهایی است که میتوان در برنامه با آن شئ انجام داد.
- اشیاء دو نوع اند:
  - Scalar : به قطعات کوچکتر تقسیم نمی شود
  - non-scalar: دارای ساختار درونی است که میتوان به آن دسترسی پیدا کرد

7

SCALAR OBJECTS

int ❑
- ✓ نماینده ی اعداد صحیح، برای مثال `5`

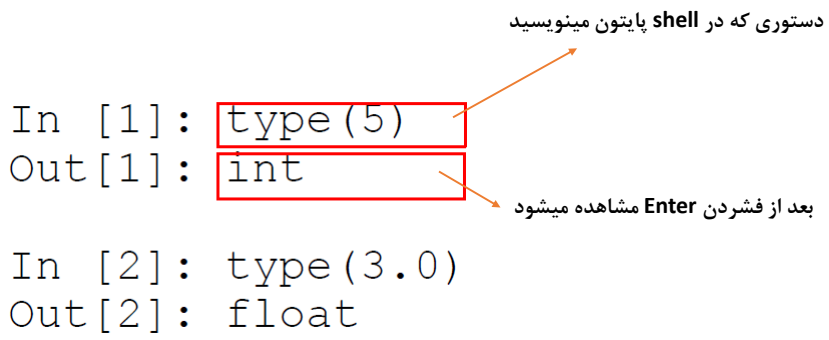
float ❑
- ✓ نماینده ی اعداد حقیقی، برای مثال `4.26`

bool ❑
- ✓ نماینده ی مقادیر منطقی شامل `True` و `False` است

NoneType ❑
- ✓ یک نوع منحصر به فرد که تنها یک مقدار دارد؛ `None`

8

با استفاده از `type()` میتوان نوع شئ را تشخیص داد.



9

TYPE CONVERSIONS (CAST)

- can **convert object of one type to another**
- `float(3)` converts integer `3` to float `3.0`
- `int(3.9)` truncates float `3.9` to integer `3`

10

PRINTING TO CONSOLE

- To show output from code to a user, use `print` command

```
In [11]: 3+2
Out[11]: 5

In [12]: print(3+2)
5
```

no 'Out' because no value returned, just something printed

11

عملیات در پایتون

- ✓ریاضی
- ✓منطقی
- ✓مقایسه ای
- ✓انتساب

12

OPERATORS ON ints and floats

- `i+j` → the **sum**
- `i-j` → the **difference**
- `i*j` → the **product**

- if both are ints, result is int
- if either or both are floats, result is float

- `i/j` → **division** - result is float
- `i//j` → **int division** - result is int, quotient without remainder
- `i%j` → the **remainder** when `i` is divided by `j`
- `i**j` → `i` to the **power** of `j`

13

SIMPLE OPERATIONS

- parentheses used to tell Python to do these operations first
  - 3*5+1 evaluates to 16
  - 3*(5+1) evaluates to 18
- **operator precedence** without parentheses
  - **
  - *
  - /
  - + and – executed left to right, as appear in expression

14

BINDING VARIABLES AND VALUES

- equal sign is an **assignment** of a value to a variable name

variable value

```
pi = 3.14159
pi_approx = 22/7
```

If use 22//7, value of expression is 3

- value stored in computer memory
- an assignment binds name to value
- retrieve value associated with name or variable by invoking the name, by typing `pi`

15

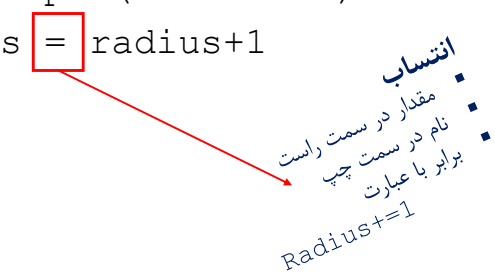
ABSTRACTING EXPRESSIONS

- why **give names** to values of expressions?
- **reuse names** instead of values
- easier to change code later

```
pi = 3.14159
radius = 2.2
area = pi*(radius**2)
```

16

```
pi = 3.14159
radius = 2.2
# area of circle
area = pi*(radius**2)
radius = radius+1
```



17

COMPARISON OPERATORS ON `int` and `float`

- `i` and `j` are any variable names

`i>j`

`i>=j`

`i<j`

`i<=j`

`i==j` → **equality** test, `True` if `i` equals `j`

`i!=j` → **inequality** test, `True` if `i` not equal to `j`

18

LOGIC OPERATORS ON bools

- `a` and `b` are any variable names

`not a` → `True` if `a` is `False`
`False` if `a` is `True`

`a and b` → `True` if both are `True`

`a or b` → `True` if either or both are `True`

19