

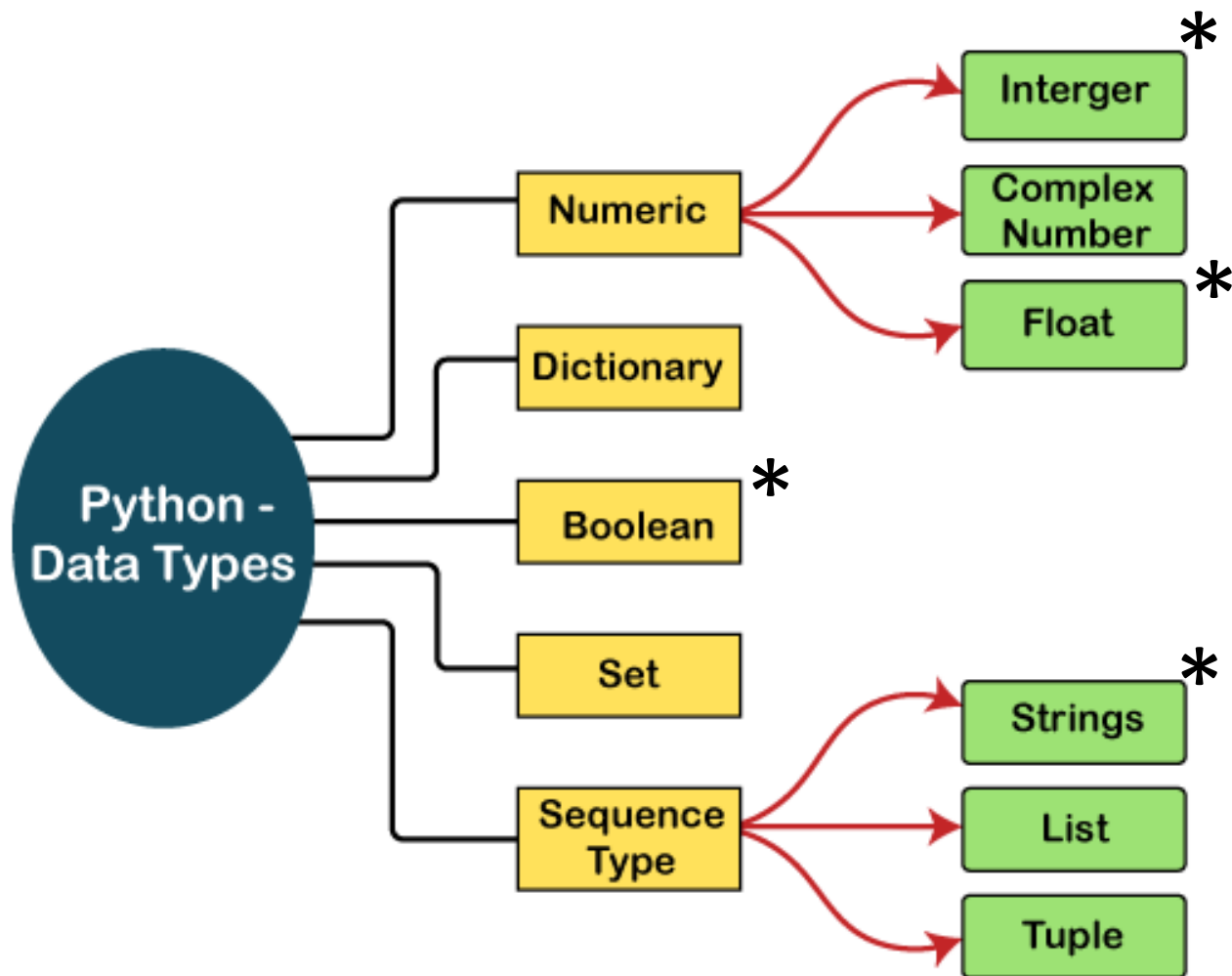
برنامه نویسی پایتون

جلسه دوم: شروع برنامه نویسی

زهرا نریمانی

دانشگاه تحصیلات تکمیلی علوم پایه زنجان، پاییز ۱۴۰۱

انواع داده اولیه در پایتون



نوع داده یا Data type

- یک مجموعه از مقادیر به همراه عملیات تعریف شده بر روی آن

<i>type</i>	<i>set of values</i>	<i>common operators</i>	<i>sample literals</i>
int	<i>integers</i>	+ - * // % **	99 12 2147483647
float	<i>floating-point numbers</i>	+ - * / **	3.14 2.5 6.022e23
bool	<i>true-false values</i>	and or not	True False
str	<i>sequences of characters</i>	+	'AB' 'Hello' '2.5'

Basic built-in data types

Overloading + operator

concatenate یا الحاق

اشیاء

- تمامی انواع داده ای در پایتون، توسط یک شیء مشخص می شوند
- با تعریف یک شیء، قسمت مشخصی از حافظه که طبق تعریف آن نوع داده تعریف شده است، اختصاص داده می شود
- هر شیء دارای یک مکان در حافظه، یک نوع داده ای (که بیان کننده رفتار شیء، یعنی مقادیر ممکن و عملیات است)، و مقدار است.

تعریف یک شیء

- نام شیء در واقع یک رفرنس یا ارجاع است که به مکان آن در حافظه اشاره می کند (شبيه اشاره گر ها در C)
- با دیدن ۱۲۳۴، اول یک متغیر از نوع صحیح ایجاد می شود و سپس به متغیر a، bind می شود.
- دسترسی به اشیاء از طریق ارجاعات (نام متغیر) انجام می شود.

لیترال

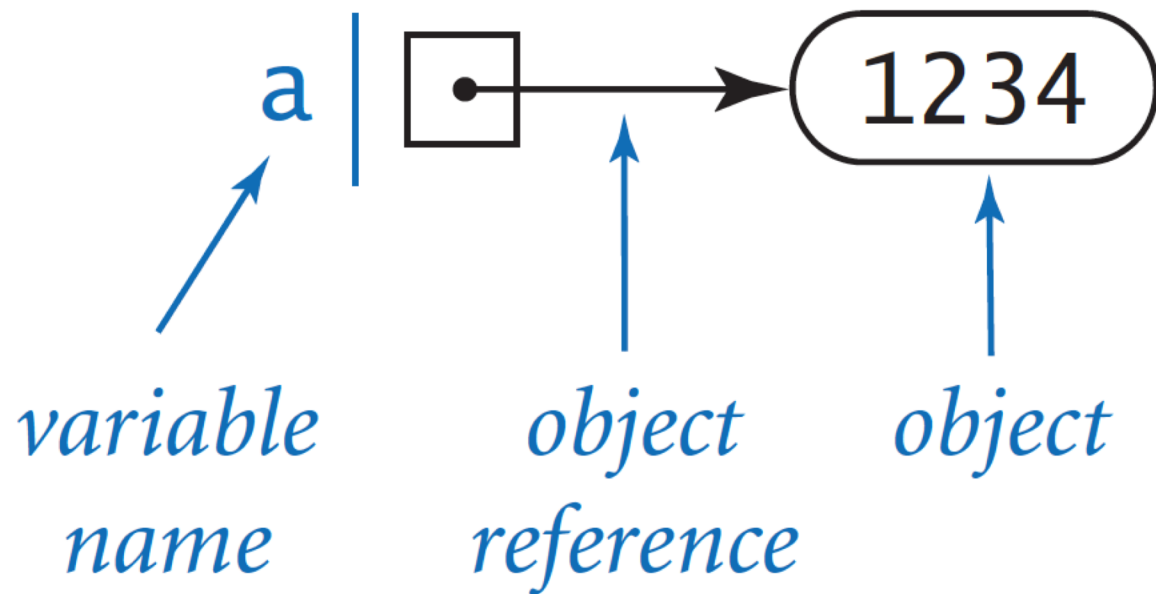
متغیر

- $a = 1234$
- $b = 99$
- $c = a + b$

سیستم مدیریت خودکار حافظه
memory error??

دستور انتساب: مقید کردن یک متغیر به یک شیء

- $a = 1234$



A variable refers to an object

عبارت یا expression

- ترکیبی از متغیرها و لیترالها و عملگرها که به نحو درستی ترکیب شده اند
- عملگر، عملوند

$$2 + (3 * X)$$

عملیات ردیابی یا trace کردن برای پیدا کردن اشکال های برنامه (دستی یا با کمک نرم افزار)

	a	b	c
a = 1234	1234		
b = 99	1234	99	
c = a + b	1234	99	1333

Your first informal trace

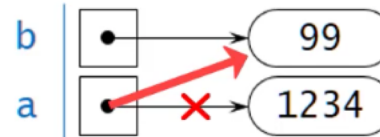
تفاوت نوع داده اولیه در پایتون و زبانهای دیگر مثل جاوا و C

- ردیابی در سطح شیء

a = 1234



b = 99



نوع داده رشته ای

- ورودی و خروجی در پایتون به صورت رشته است
- نمی توان یک عدد را به صورت یک عدد خواند!
- همچنین اهمیت رشته ها در برنامه های پردازش متن
- تفاوت فضای خالی با فاصله در داخل رشته

'Hi, ' + 'Mary'

'1234' + ' ' + '10'

یک مثال



The ruler function for $n = 4$

Program 1.2.1 String concatenation example (ruler.py)

```
import stdio

ruler1 = '1'
ruler2 = ruler1 + ' 2 ' + ruler1
ruler3 = ruler2 + ' 3 ' + ruler2
ruler4 = ruler3 + ' 4 ' + ruler3
stdio.writeln(ruler1)
stdio.writeln(ruler2)
stdio.writeln(ruler3)
stdio.writeln(ruler4)
```

% python ruler.py

```
1
1 2 1
1 2 1 3 1 2 1
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1
```

This program writes the relative lengths of the subdivisions on a ruler. The n th line of output is the relative lengths of the marks on a ruler subdivided in intervals of $1/2^n$ of an inch.

تبدیل نوع یا type casting

Print vs. print

- تبدیل عدد به رشته برای خروجی:

```
print(str(a) + " + " + str(b) + " = " + str(c))
```

دریافت ورودی

```
1 import sys
2
3 print( 'Hi, ' + sys.argv[1] + '!.')
4 print( 'Bye!')
5
```

اجرا

```
$ python arg.py Mary
Hi, Mary.
Bye!
```

```
1 import sys
2
3 a = int(sys.argv[1])
4 b = int(sys.argv[2])
5
6 print(a+b)
7
```

اجرا

???

پیاده سازی اولین فلوچارت ...

انواع داده عددی در پایتون

<i>type</i>	<i>set of values</i>	<i>common operators</i>	<i>sample literals</i>
int	<i>integers</i>	+ - * // % **	99 12 2147483647
float	<i>floating-point numbers</i>	+ - * / **	3.14 2.5 6.022e23
bool	<i>true-false values</i>	and or not	True False
str	<i>sequences of characters</i>	+	'AB' 'Hello' '2.5'

Basic built-in data types

نوع عدد صحیح یا integer

<i>values</i>	<i>integers</i>						
<i>typical literals</i>	1234 99 0 1000000						
<i>operations</i>	<i>sign</i>	<i>add</i>	<i>subtract</i>	<i>multiply</i>	<i>floored divide</i>	<i>remainder</i>	<i>power</i>
<i>operators</i>	+ -	+	-	*	//	%	**

Python's int data type

<i>expression</i>	<i>value</i>	<i>comment</i>
99	99	<i>integer literal</i>
+99	99	<i>positive sign</i>
-99	-99	<i>negative sign</i>
5 + 3	8	<i>addition</i>
5 - 3	2	<i>subtraction</i>
5 * 3	15	<i>multiplication</i>
5 // 3	1	<i>no fractional part</i>
5 % 3	2	<i>remainder</i>
5 ** 3	125	<i>exponentiation</i>
5 // 0	<i>run-time error</i>	<i>division by zero</i>
3 * 5 - 2	13	<i>* has precedence</i>
3 + 5 // 2	5	<i>// has precedence</i>
3 - 5 - 2	-4	<i>left associative</i>
(3 - 5) - 2	-4	<i>better style</i>
3 - (5 - 2)	0	<i>unambiguous</i>
2 ** 2 ** 3	256	<i>right associative</i>
2 ** 1000	107150...376	<i>arbitrarily large</i>

خطا

Typical int expressions

توجه به خطاها (معمولاً آخرین خط خروجی مهم است)

```
[>>> 1 // 0
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: integer division or modulo by zero
```

نام خطا

توضیح علت خطا

تمرین: آیا موفق اجرا می شود؟

```
1  import sys
2
3  a = int(sys.argv[1])
4  b = int(sys.argv[2])
5
6  total = a + b
7
8  print(a + ' + ' + b + ' = ' + total)
9
```