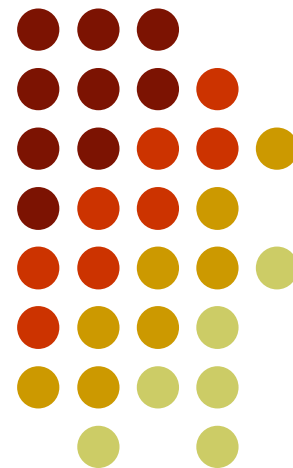


مبانی کامپیوتر و برنامه سازی

درس سوم: زبان برنامه نویسی پایتون

مدرس: علی فانیان





- پایتون یک زبان برنامه نویسی با مشخصات
 - همه منظوره
 - متن باز
 - شی گرا
 - مفسری
 - دارای توان بالا با قواعد دستوری بسیار ساده
 - قابل استفاده در سیستم عامل های مختلف
- هزاران برنامه نویس در دنیا که از این زبان به منظور
 - پیاده سازی اسکریپت ها
 - برنامه های سیستمی
 - برنامه های واسطه کاربری



- ارائه آن در سال ۱۹۹۰ توسط Guido van Rossum
- در دسترس بودن توابع کتابخانه (library)
- پشتیبانی از مدل برنامه نویسی ماژولار
- در اختیار بودن هزاران ماژول توسعه داده شده برای توسعه برنامه جدید
- استفاده غول های IT دنیا از این زبان در محصولات خود
- Google ، YouTube ، Facebook و
- استفاده گسترده آن در هوش مصنوعی
- بسیار مناسب برای افرادی که تازه برنامه نویسی را شروع می کنند



نصب پایتون

- وجود دو نسخه ۲ و ۳ در پایتون
- آخرین نسخه پایتون ۳.۱۰
- در سایت python.org می توان پایتون را برای سیستم عامل های مختلف دانلود و نصب نمود
- روند نصب پایتون در ویندوز در ادامه
 - افزودن مسیر نصب پایتون در path
- استفاده از Jupyter Notebook



انواع دیتاتایپ ها در پایتون

- تایپهای عددی در پایتون

`x = 1 # int`

- `int`: Integer و یا عدد صحیح

- در پایتون `int` محدودیت روی تعداد ارقام ندارد

`y = 2.8 # float`

- `float`: عدد اعشاری

`z = 1j # complex`

- `complex`:

- `bool`: منطقی یا بولین دارای دو مقدار `True`, `False` می تواند

`b = 12 == 13`

باشد



انواع دیتاتایپ ها در پایتون

- **Str:** متن و یا رشته. برای بیان رشته باید عبارت را داخل علامت quotation و یا double quotation قرار داد

- در برخی از زبانها کارکتر و رشته متفاوت در نظر گرفته می شود در پایتون نه
`s='hello' or s="hello"`

- **List:** لیست ها برای ذخیره داده های مختلف در یک متغیر. داده هایی که حتی انواع آنها متفاوت است و می تواند در لیست حضور داشته باشد مثلا نام، سن، معدل و

```
list1 = ["apple", "banana", "cherry"]
```

```
list2 = [1, 5, 7, 9, 3]
```

```
list3 = [True, False, False]
```

```
list1 = ["abc", 34, True, 40, "male"]
```



انواع دیتاتایپ ها در پایتون

- Dict: این تایپ برای ذخیره داده های بصورت key:value بکار می روند. این تایپ ordered است و عناصر آن قابل تغییر و امکان عنصر تکراری ندارد

```
thisdict = {"brand": "Ford", "model": "Mustang", "year": 1964}  
print(thisdict["brand"])
```

- Tup یا tuple: ذخیره داده های مختلف در یک متغیر مانند لیست با این تفاوت که قابل تغییر نیست مقادیر آنها

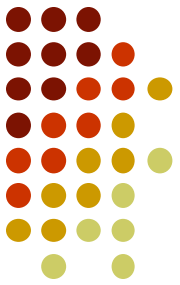
```
thistuple = ("apple", "banana", "cherry")  
print(thistuple[1])
```

عملگرها در پایتون



- عملگرهای ریاضی (Arithmetic operators)
- عملگرهای انتصاب (Assignment operators)
- عملگرهای مقایسه ای (Comparison operators)
- عملگرهای منطقی (Logical operators)
- عملگرهای شناسه ای (Identity operators)
- عملگرهای عضویت (Membership operators)
- عملگرهای بیتی (Bitwise operators)

عملگرهای ریاضی



Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	<u>$x // y$</u>

عملگرهای بیتی



Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1
	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off
>>	Signed right shift	<u>Shift right by pushing copies of the leftmost bit in from the left</u> , and let the rightmost bits fall off



عملگرهای انتصاب

Operator	Example	Same As	توضیحات
=	x = 5	x = 5	
+=	x += 3	x = x + 3	
-=	x -= 3	x = x - 3	
*=	x *= 3	x = x * 3	
/=	x /= 3	x = x / 3	تقسیم اعشاری
%=	x %= 3	x = x % 3	باقیمانده تقسیم
//=	x //= 3	x = x // 3	تقسیم بصورت عدد صحیح(حذف قسمت اعشار)
**=	x **= 3	x = x ** 3	به توان رسانی
&=	x &= 3	x = x & 3	ترکیب عطفی بیتها . در صورتی خروجی یک است که هر دو بیت یک باشند
=	x = 3	x = x 3	ترکیب فصلی بیتها . در صورتی خروجی یک است که حداقل یک بیت یک باشند
^=	x ^= 3	x = x ^ 3	ترکیب فصلی امحصاری بیتها . در صورتی خروجی یک است که هر دو بیت متفاوت باشند
>>=	x >>= 3	x = x >> 3	شیفت به راست دادن بیتها به تعداد مشخص شده
<<=	x <<= 3	x = x << 3	شیفت به سمت چپ دادن بیتها به تعداد مشخص شده

عملگرهای مقایسه ای



Operator	Name	Example	توضیحات
==	Equal	$x == y$	در صورت برقرار بودن شرط
!=	Not equal	$x != y$	مقدار True باز می گردد و
<	Greater than	$x > y$	در غیر این صورت مقدار false
>	Less than	$x < y$	
>=	Greater than or equal to	$x >= y$	
<=	Less than or equal to	<u>$x <= y$</u>	



عملگرهای منطقی

- برای ساختن ترکیب های شرطی مختلف: ورودی این عملگرها Boolean هستند

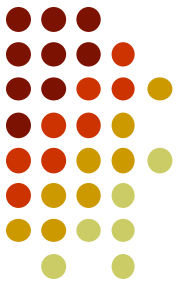
Operator	Description	Example	توضیحات
and	Returns True if both statements are true	$x < 5$ and $x < 10$	خروجی مقدار True
or	Returns True if one of the statements is true	$x < 5$ or $x < 4$	خواهد داشت اگر شرط برقرار باشد
not	Reverse the result, returns False if the result is true	$\text{not}(x < 5 \text{ and } x < 10)$	<u>در غیر این صورت مقدار False دارد</u>

عملگرهای عضویت



Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	x in y
not in	Returns True if a sequence with the specified value is not present in the object	<u>x not in y</u>

اولویت عملگرها



Operator	Description
**	Exponentiation (raise to the power)
~ + -	Complement, unary plus and minus (method names for the last two are +@ and -@)
* / % //	Multiply, divide, modulo and floor division
+ -	Addition and subtraction
>> <<	Right and left bitwise shift
&	Bitwise 'AND'
^	Bitwise exclusive 'OR' and regular 'OR'
<= < > >=	Comparison operators
<> == !=	Equality operators
= %= /= //= -= += *= **=	Assignment operators
is is not	Identity operators
in not in	Membership operators
not or and	Logical operators



قواعد نام گذاری متغیرها

- متغیر با یک حرف و یا کارکتر `underline` شروع می شود
- متغیر نمی تواند با عدد شروع شود
- متغیر می تواند فقط شامل کارکترهای حرفی، عددی و `underline` باشد
- متغیرها اصلاً `case sensitive` هستند یعنی حساس به حروف کوچک بزرگ `age, Age, AGE` سه متغیر متفاوت است
- از کلمات کلید زبان استفاده نکنیم مانند `list, str` و...



انواع متغیرها

- پایتون زبان dynamic typing است یعنی یک متغیر را می توان در طول برنامه به صورت های مختلف بکار برد

`X=2`

`X="How are you"`

`X=[1,2,3]`

- برخلاف بسیاری از زبانهای دیگر مانند C که static typing است یعنی اگر متغیر `int` تعریف شد در طول برنامه `type` آن تغییر نمی کند



انواع متغیرها

- اگر متغیر مقدار دهی نشده باشد و استفاده شود `name error` می دهد
- با توجه به این که تایپ متغیر ممکن است عوض شود تابع `type` نوع متغیر را در وضعیت جاری می دهد

```
a = ('apple', 'banana', 'cherry')  
b = "Hello World"  
c = 33
```

```
x = type(a)  
y = type(b)  
z = type(c)
```

```
print(type([]) is list)  
  
print(type([]) is not list)  
  
print(type(()) is tuple)  
  
print(type({}) is dict)  
  
print(type({}) is not list)
```

Output:

```
True  
False  
True  
True  
True
```

برخی دستورات مهم در پایتون



- If ... Else
- While Loops
- For Loops
- User Input

قواعد کدنویسی در پایتون: تورفتگی خطوط پایتون Indentatio



```
Statement
if condition:
    if condition:
        Statement
    else:
        Statement
Statement
```

نحوه‌ی خوانش مفسر

```
شروع بلوک اول کد
ادامه‌ی بلوک اول کد
    شروع بلوک دوم کد
        شروع بلوک سوم کد
        ادامه‌ی بلوک دوم کد
        ادامه‌ی بلوک سوم کد
    ادامه‌ی بلوک اول کد
```



معرفی string ها

- String ها دنباله ای از کاراکترها هستند
..., "ali", "hello"
- با استفاده از تابع len می توان طول string را بدست آورد
- باتوجه به این که string ها ordered هستند می توان با slicing و یا Indexing به عناصر آن ها دست یافت
- برای indexing از [] استفاده می کنیم
- با indexing می توان یک کاراکتر را از string برداشت

Characters		h	e	l	l	o
Index	0	1	2	3	4	
Reverse index		0	-4	-3	-2	-1

این کارکتر n ام را از آخر می دهد



معرفی string ها

- Slicing یعنی برداشت یک subsection یا زیر بخش

My_string[start:stop:step]

اندیس شروع

اندیس پایانی خودش را شامل نمی شود

گام (چند تا)



Method	Description
<u>capitalize()</u>	Converts the first character to upper case
<u>count()</u>	Returns the number of times a specified value occurs in a string
<u>index()</u>	Searches the string for a specified value and returns the position of where it was found
<u>find()</u>	Searches the string for a specified value and returns the position of where it was found
<u>isdigit()</u>	Returns True if all characters in the string are digits
<u>split()</u>	Splits the string at the specified separator, and returns a list
<u>islower()</u>	Returns True if all characters in the string are lower case
<u>±</u>	To concatenate, or combine, two strings you can use the+ operator
<u>format()</u>	Formats specified values in a string
<u>Escape characters</u>	To insert characters that are illegal in a string
<u>strip()</u>	Returns a trimmed version of the string
<u>replace()</u>	Returns a string where a specified value is replaced with a specified value



لیست ها در پایتون

- لیست ها دنباله های منظمی هستند که می توانند تایپ های مختلفی را به خود بگیرند
- لیست های اصطلاحاً `ordered` هستند یعنی هر عنصری دارای جایگاه مشخصی و بر اساس مکانی که وارد آن شده تغییر نمی کند
- اگر عنصر جدیدی به لیست افزوده شود به انتهای لیست اضافه می شود.
- امکان تغییر یک عنصر و یا حذف آن نیز وجود دارد
- مقادیر مشابه می تواند در لیست باشد
- لیست ها امکان `slicing` و `indexing` و متدهای مختلفی دارند
- `len`
- `append`

لیست در پایتون



- Access List Items
- Change List Items
- Add List Items
- Remove List Items
- Loop Lists
- Sort Lists
- Copy Lists
- Join Lists

متدوها در لیست



Method	Description
<u>append()</u>	Adds an element at the end of the list
<u>clear()</u>	Removes all the elements from the list
<u>copy()</u>	Returns a copy of the list
<u>count()</u>	Returns the number of elements with the specified value
<u>extend()</u>	Add the elements of a list (or any iterable), to the end of the current list
<u>index()</u>	Returns the index of the first element with the specified value
<u>insert()</u>	Adds an element at the specified position
<u>pop()</u>	Removes the element at the specified position
<u>remove()</u>	Removes the item with the specified value
<u>reverse()</u>	Reverses the order of the list
<u>sort()</u>	Sorts the list