



Data Types: انواع داده

داده های پرکاربرد		
عملیات	کد	مثال
عدد صحیح	<code>int(number)</code>	10
عدد اعشاری	<code>float(number)</code>	10.1
بولین	<code>bool(input)</code>	True , False
رشته	<code>str(text)</code>	"Hello World"
لیست	<code>list(arragments)</code>	[0, 1, 2, 3, 4]
تاپل	<code>(value1, value2, ...)</code>	
دیکشنری	<code>dict(arragments)</code>	{'one': 1, 'two': 2, 'three': 3}
توالی	<code>range(start, stop, step)</code> <code>range(start, stop)</code> <code>range(stop)</code>	1, 3, 5, 7, 9 2, 3, 4, 5, 6 0, 1, 2, 3, 4

داده نوع ۱: عدد صحیح

```
# int: Convert a string or float to an integer
num_str = "10"
num_float = 10.5
int_num_str = int(num_str)
int_num_float = int(num_float)
print(int_num_str) # Output: 10
print(int_num_float) # Output: 10
```

`num_float` تبدیل به عدد صحیح میشود و از قسمت اعشاری صرف نظر می شود .

داده نوع ۲: عدد اعشاری

```
# float: Convert a string or integer to a float
num_str = "10.5"
num_int = 10
float_num_str = float(num_str)
float_num_int = float(num_int)
print(float_num_str) # Output: 10.5
print(float_num_int) # Output: 10.0
```

`num_int` تبدیل به عدد اعشار میشود و به قسمت اعشاری آن اضافه می شود .





داده نوع ۳ : بولین

```
# bool: Convert different types to a boolean
user_input = input("Enter something: ")
bool_input = bool(user_input)
print(bool_input) # Output: True if the user enters anything non-empty, False otherwise
```

در صورتی که فرد در ورودی چیزی وارد کند یا عددی غیر از صفر وارد کند نتیجه True است .

داده نوع ۵ : رشته

```
# str: Convert numbers to a string
number = 123
text = "The number is: " + str(number)
print(text) # Output: The number is: 123
```

در صورتی که فرد عدد را تبدیل به رشته نکند ، نمی تواند به متن ، عدد صحیح را اضافه کند و با ارور مواجه می شود .

داده نوع ۶ : لیست

```
# list: Convert a range to a list
range_to_list = list(range(5))
print(range_to_list) # Output: [0, 1, 2, 3, 4]
```

* برای اضافه کردن به لیست از `append` استفاده می کنیم .

```
# First, define a list
my_list = [1, 2, 3, 4, 5]

# Adding an element to the list using append
my_list.append(6)
print("After appending 6:", my_list) # Output: [1, 2, 3, 4, 5, 6]
```

* برای حذف کردن از لیست از دستور `remove` استفاده می کنیم .

```
my_list.remove(3)
print("After removing 3:", my_list) # Output: [1, 2, 4, 5, 6]
```

برای جایگذاری کردن در لیست از عدد جایگاه `index` استفاده می کنیم .

```
# Replacing an element in the list
my_list[2] = 10 # Replace the value 4 with 10
print("After replacing 4 with 10:", my_list) # Output: [1, 2, 10, 5, 6]
```





برای شمارش و چاپ از جایگاه باید طبق روش زیر پیش برویم :

[گام : پایان : شروع]

۰	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳
T	h	i	s		i	s		a		t	e	x	t
-۱۴	-۱۳	-۱۲	-۱۱	-۱۰	-۹	-۸	-۷	-۶	-۵	-۴	-۳	-۲	-۱

مثلا :

```
# Define the text string
text = "This is a text"

# Print the substring from index 0 to 3 (excluding 4)
print(text[0:4]) # Output: This

# Print the string in reverse using slicing with a step of -1
print(text[::-1]) # Output: txet a si sihT
```

داده نوع ۷ : تاپل

```
# Defining a tuple of fruits
fruits = ("apple", "banana", "cherry", "date")

# Accessing elements of a tuple
print(fruits[0]) # Output: apple
print(fruits[1]) # Output: banana

# Iterating through a tuple
for fruit in fruits:
    print(fruit)

# Attempting to modify a tuple (this will cause an error)
# fruits[0] = "orange" # TypeError: 'tuple' object does not support item assignment
```

داده های این نوع داده تغییر ناپذیر است و قابل تغییر نیست .





داده نوع ۸ : دیکشنری

```
# Create a dictionary with student details using the dict() function
student_dict = dict(name="Amirreza Jafari", student_id="123456", gpa=3.5)

# Print the original dictionary
print("Original dictionary:", student_dict)
# Output: {'name': 'Amirreza Jafari', 'student_id': '123456', 'gpa': 3.5}
```

نحوه اضافه کردن عنصر :

```
# Add an element to the dictionary
student_dict["major"] = "Computer Engineering"
print("After adding 'major':", student_dict)
# Output: {'name': 'Amirreza Jafari', 'student_id': '123456', 'gpa': 3.5, 'major': 'Computer Engineering'}
```

* برای حذف کردن عنصر باید از del استفاده می کنیم .

```
# Remove an element from the dictionary
del student_dict["gpa"]
print("After removing 'gpa':", student_dict)
# Output: {'name': 'Amirreza Jafari', 'student_id': '123456', 'major': 'Computer Engineering'}
```

برای جایگذاری کردن در دیکشنری از نام عنصر جایگاه استفاده می کنیم .

```
# Replace an element in the dictionary
student_dict["name"] = "Alireza Jafari"
print("After replacing 'name':", student_dict)
# Output: {'name': 'Alireza Jafari', 'student_id': '123456', 'major': 'Computer Engineering'}
```





داده نوع ۹ : توالی

```
# Example of converting range to a list
numbers_list = list(range(1, 10, 2))
print(numbers_list)

# range with start, stop, and step
for i in range(1, 10, 2):
    print(i)
# Output: 1, 3, 5, 7, 9

# range with start and stop
for i in range(2, 7):
    print(i)
# Output: 2, 3, 4, 5, 6

# range with stop
for i in range(5):
    print(i)
# Output: 0, 1, 2, 3, 4
```

stop : مقدار پایان است و نوشتنش اجبار است و پیش فرضی ندارد :

start : مقدار شروع است و به صورت پیش فرض روی صفر است :

step : گام شمارش است و به صورت پیش فرض روی یک است :

* داده **توالی** استفاده زیادی در حلقه **for** دارد زیرا باعث ایجاد حلقه میشود . *

