



DOKUZ EYLÜL
ÜNİVERSİTESİ
İSTATİSTİK
BÖLÜMÜ

Computer Programming – 2

Alper VAHAPLAR

2019 – 2020

General Information of the Course

Course Code	BIL3120
Course Name	Computer Programming – 2
Course Level	Undergraduate
Course Status	Elective
Language of Instruction	English
Weekly Course Hours	3+0 (Theory + Application)
National Credit	3+0
ECTS Credit	5

General Information of the Course

Course Code	BIL3120
Course Name	Computer Programming – 2
Course Days	Wednesday, 2, 3, 4 th hours (09:25 – 12:00)
Classroom	Comp. Lab. – 1
Course Web Page	http://alpervahaplar.com – BIL3120

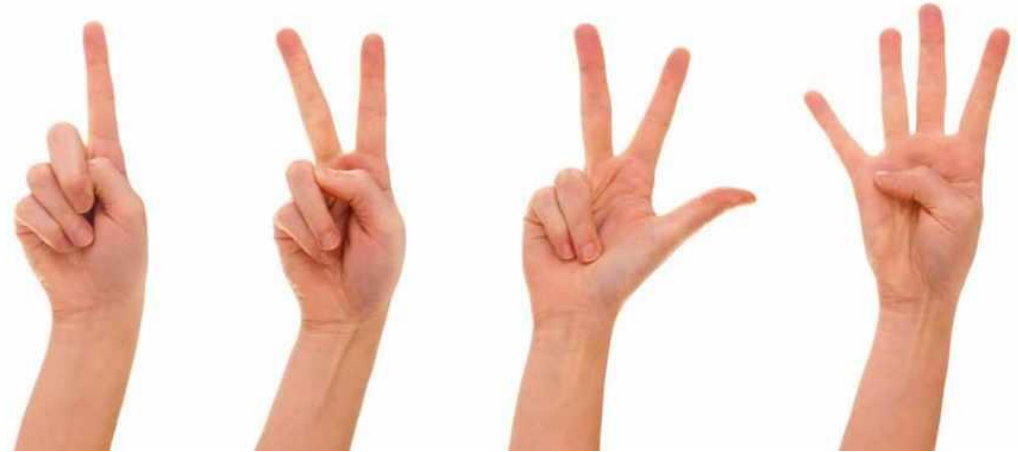
Course Evaluation

- 1 Midterm Exam (MTE)
- Assignments (ASG)
 - ✓ Homeworks
 - ✓ Quizes
- Final Exam (FIN)
- **Score = MTE * 0.40 + ASG * 0.10 + FIN * 0.50**

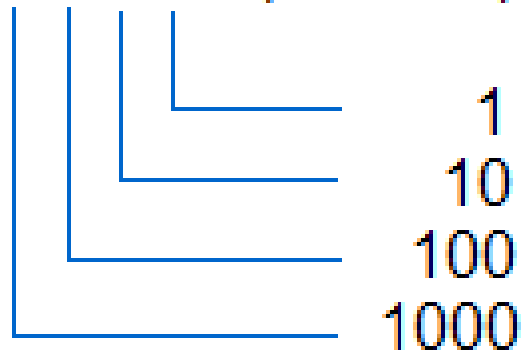
Introduction: *"Number Systems"*

Number Systems – Human

- We (humans) use "Decimal" system.
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Why?

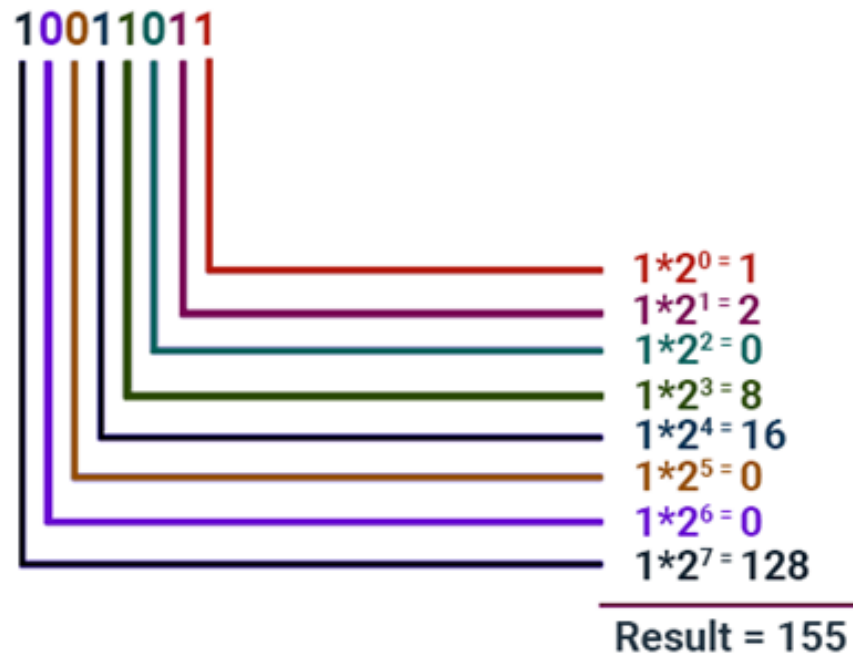


$$9\ 5\ 4\ 8 = (9 \times 1000) + (5 \times 100) + (4 \times 10) + (8 \times 1)$$



Number Systems – Computer

- Which system do "computers" use?
- Binary System (0, 1)
- Binary Numbers are the flow of information in the form of zeros and ones used by digital computers and systems.
- Why?



Number Systems

System	Base	Symbols	Used by humans?	Used in computers?
Decimal	10	0, 1, ... 9	Yes	No
Binary	2	0, 1	No	Yes
Octal	8	0, 1, ... 7	No	No
Hexa-decimal	16	0, 1, ... 9, A, B, ... F	No	Sometimes

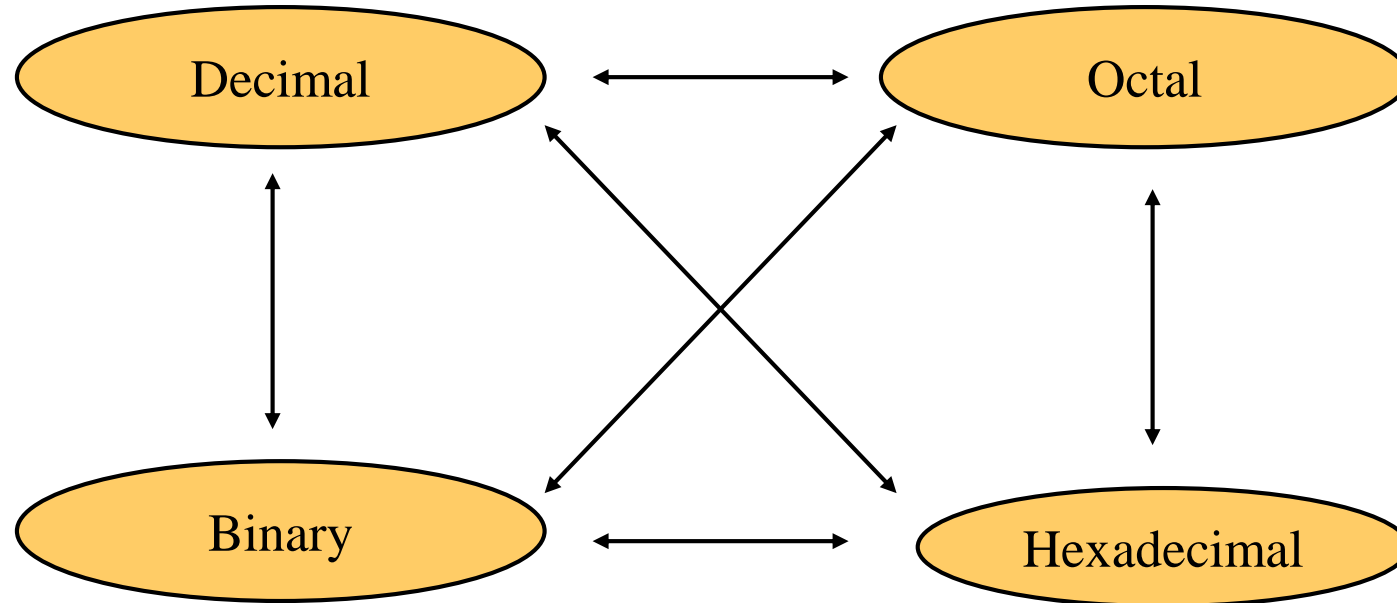
Quantities/Counting

Decimal	Binary	Octal	Hexa- decimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7

Decimal	Binary	Octal	Hexa- decimal
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Conversion Among Bases

➤ The possibilities:



Exercise – Convert ...

Decimal	Binary	Octal	Hexa- decimal
33			
	1110101		
		703	
			1AF

Exercise – Convert ...

Decimal	Binary	Octal	Hexa- decimal
33	100001	41	21
117	1110101	165	75
451	111000011	703	1C3
431	110101111	657	1AF

Algorithm

Algorithm

Algorithm, named after the 9th century scholar Ebu Abdullah Muhammed bin Musa El-Harezmi.



An algorithm is;

- any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output.
- an algorithm is thus a sequence of computational steps that transform the input into the output.

Expressing Algorithms

➤ Step Form

- ✓ sequence of numbered steps or points

➤ Pseudo code

- ✓ generic way of describing an algorithm without use of any programming language syntax

➤ Flow Charts

- ✓ graphical representation of an algorithm

Expressing Algorithms

➤ Step Form

- ✓ sequence of numbered steps or points

➤ Example: Write an algorithm to find the sum of 2 numbers:

- 1. START
- 2. Read the first number
- 3. Read the second number
- 4. Find the sum of the two numbers
- 5. Print the sum of the numbers
- 6. STOP.

Expressing Algorithms

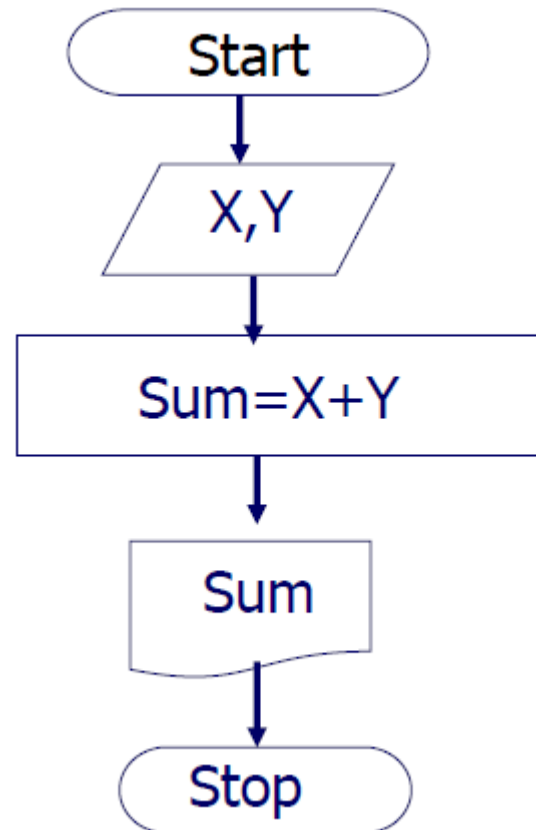
- Pseudo code
 - ✓ generic way of describing an algorithm without use of any programming language syntax.
- Example: Write an algorithm to find the sum of 2 numbers:
 - 1. START
 - 2. Read X
 - 3. Read Y
 - 4. $\text{Sum} \leftarrow X+Y$
 - 5. Print Sum
 - 6. STOP.

Expressing Algorithms

➤ Flow Charts

✓ graphical representation of an algorithm

➤ Example: Write an algorithm to find the sum of 2 numbers:





Python Programming Language

Python Programming Language



➤ Python

- ✓ is a **general purpose, interpreted** programming language.
- ✓ is a language that supports multiple approaches to software design, principally **structured** and **object-oriented** programming.
- ✓ provides automatic **memory management** and **garbage collection**.
- ✓ is **extensible**.

Python Programming Language



- Guido Van Rossum(Amsterdam, December 1989)
 - ✓ Monty Python – English Comedian Group
- Goals:
 - ✓ An easy and intuitive language just as powerful as major competitors
 - ✓ Open source, so anyone can contribute to its development
 - ✓ Code that is as understandable as plain English
 - ✓ Suitability for everyday tasks, allowing for short development times

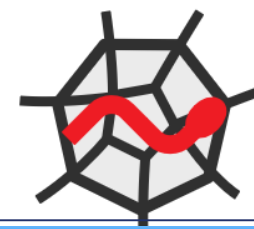
Spyder – Python Editor



➤ Scientific Python Development EnviRonment

- ✓ Written in Python, for Python
- ✓ Designed by and for scientists, engineers and data analysts.
- ✓ IDE – Integrated Development Environment

Spyder – Python Editor



The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python file with the following content:

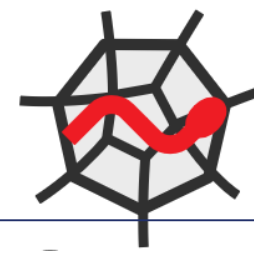
```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Oct 1 16:41:00 2019
4
5 @author: Alper
6 """
7
8
```

The Variable explorer on the right is empty, with the text "Variable / File explorer" overlaid. The IPython console at the bottom shows the Python 3.6.4 prompt:

```
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more
>>>
IPython 6.2.0 -- An enhanced Interactive Python.
>>>
In [1]:
```

The status bar at the bottom shows: Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 8, Column: 1, Memory: 52%.

Spyder – Python Editor



- Spyder Console (Ipython)
 - ✓ Command Line

```
Console 1/A [x]
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.2.1 -- An enhanced Interactive Python.

In [1]:
```


Operators in Python



➤ Arithmetic Operators

Operator	Operation	Example
+	Addition	48 + 23
-	Subtraction	48 - 23
*	Multiplication	48 * 23
/	Division	48 / 23
%	Modulus	48 % 23
**	Exponent	48 ** 23
//	Floor Division	48 // 23

Operators in Python



➤ Comparison Operators

Operator	Operation	Example
<code>==</code>	Equal	<code>48 == 23</code>
<code>!=</code>	Not Equal	<code>48 != 23</code>
<code><></code>		<code>48 <> 23</code>
<code>></code>	Greater than	<code>48 > 23</code>
<code><</code>	Smaller than	<code>48 < 23</code>
<code>>=</code>	Greater than or equal to	<code>48 >= 23</code>
<code><=</code>	Less than or equal to	<code>48 <= 23</code>

Operators in Python



➤ Compound Operators

Operator	Example	Equivalent to
+=	<code>x += 3</code>	<code>x = x + 3</code>
-=	<code>x -= 3</code>	<code>x = x - 3</code>
*=	<code>x *= 3</code>	<code>x = x * 3</code>
/=	<code>x /= 3</code>	<code>x = x / 3</code>
%=	<code>x %= 3</code>	<code>x = x % 3</code>

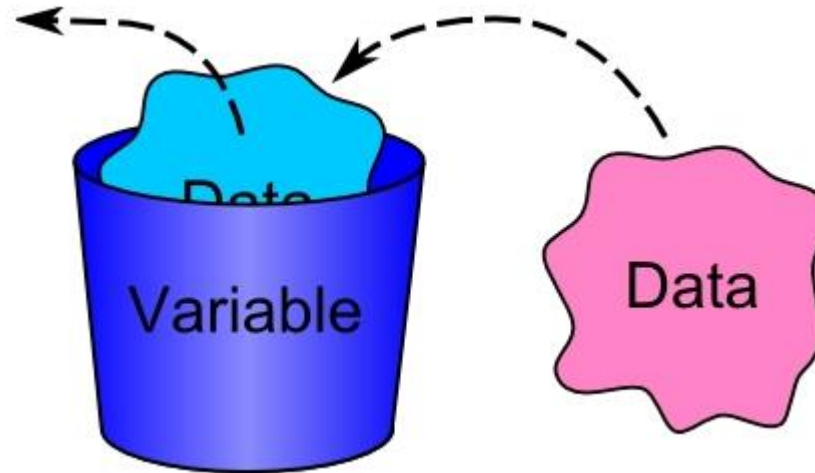
Variables in Python



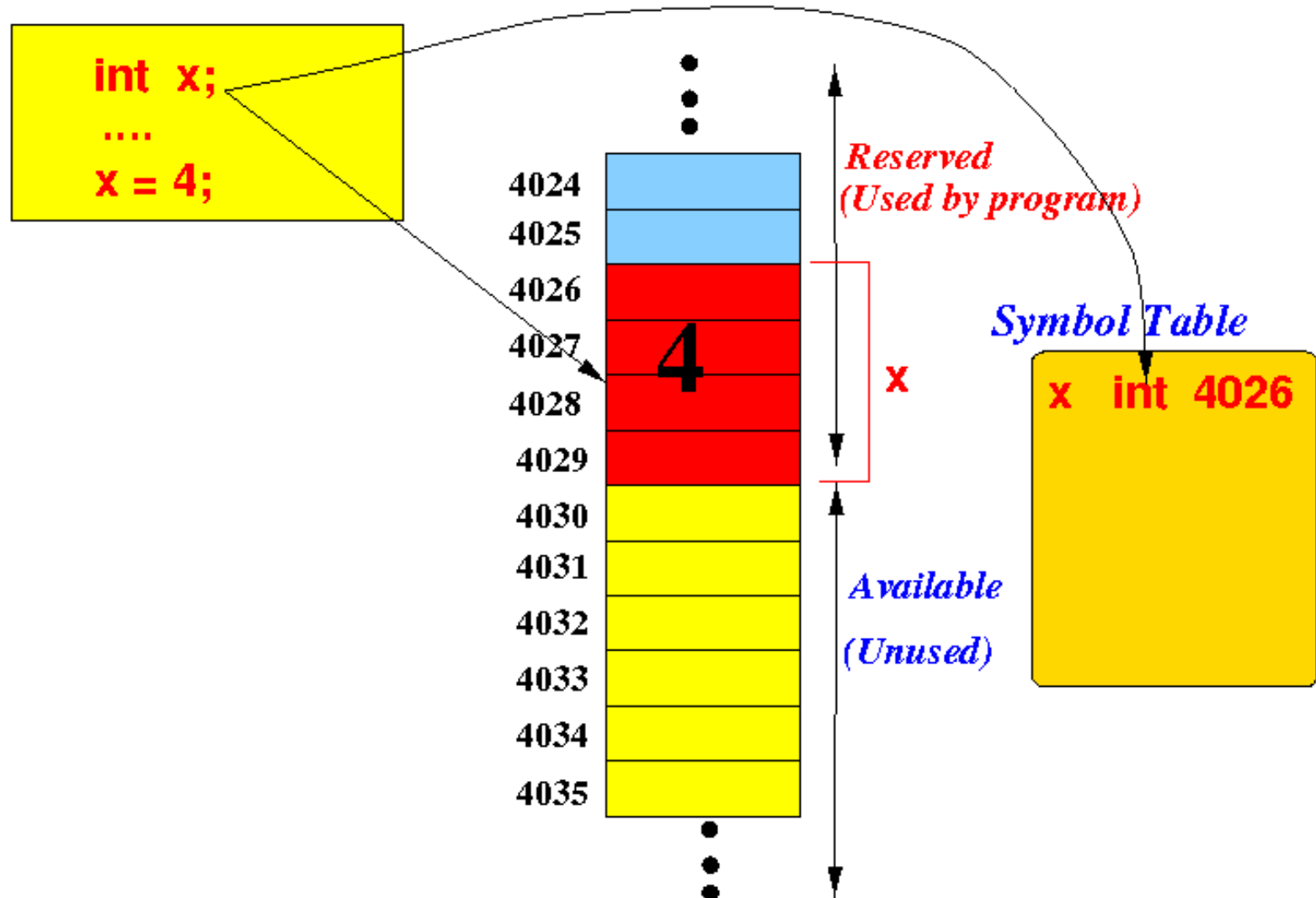
➤ A **variable** is a named memory location in which data of a certain type can be stored.

➤ A variable has :

- ✓ Name
- ✓ Address
- ✓ Type
- ✓ Value
- ✓ Scope



Variables in Python



Variables in Python



➤ Variable Name:

- ✓ Must begin with a letter (a - z, A - B) or underscore _
- ✓ Other characters can be letters, numbers or _
- ✓ Is case sensitive: capitalization counts!
- ✓ Can be any reasonable length.

➤ Ex: toplam, sayac, adet, gecme_notu, _temp
yeni_gelen_musteri_sayisi

Variables in

➤ Assignment Operat

=

➤ Ex:

- ✓ `toplam = 0`
- ✓ `adet = 12 + 75`
- ✓ `ad_soyad = "Alper VAHAPLAR"`
- ✓ `alan = 2 * 3.14 * (5**2)`

Name	Type	Size	
<code>ad_soyad</code>	<code>str</code>	1	Alper Vahaplar
<code>adet</code>	<code>int</code>	1	87
<code>alan</code>	<code>float</code>	1	157.0
<code>toplam</code>	<code>int</code>	1	0

Variable explorer File explorer Help

Variables in Python



➤ Assignment Operator: (=)

✓ `x = 3`

✓ `y = 2`

✓ `z = x + y`

✓ `x = x + 1`

➤ Assignments can be done *en masse*:

✓ `x = y = z = 5`

➤ Multiple assignments can be done on one line:

✓ `x, y, z = 44, 3.14, 'Hüseyin'`

Data Types in Python



➤ *Basic Types:*

- ✓ *Boolean (True or False)*
- ✓ *Integer Numbers (45),*
- ✓ *Floating Point Numbers (3.14),*
- ✓ *Complex Numbers (3 + 2j),*
- ✓ *Strings ("Alper", 'Computer Programming')*

Data Types in Python



- *Boolean (True or False),*
 - ✓ In [1]: `a = 45 < 23`
- *Integer Numbers (45),*
 - ✓ In [2]: `b = 45`
- *Floating Point Numbers (3.14),*
 - ✓ In [3]: `c = 6.02`
- *Complex Numbers (3 + 2j),*
 - ✓ In [4]: `d = 3 + 2j`
- *Strings ("Alper", 'Computer Programming')*
 - ✓ In [5]: `e = "Bi ara mı versek? :)"`

Data Types in Python



Name	Type	Size	
a	bool	1	False
b	int	1	45
c	float	1	6.02
d	complex	1	(3+2j)
e	str	1	Bi ara mı versek? :)

Data Types in Python



Name	Type	Size	
a	bool	1	False
b	int	1	45
c	float	1	6.02
d	complex	1	(3+2j)
e	str	1	Bi ara m1 versek? :)

```
In [48]: type(a)
Out[48]: bool
```

```
In [49]: type(b)
Out[49]: int
```

```
In [50]: type(c)
Out[50]: float
```

Data Types in Python



➤ *Other Types:*

- ✓ *Lists*

- ✓ *Dictionaries*

- ✓ *Tuples*

- ✓ *Sets*