



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

Python Data Types

HADAIQ ROLIS SANABILA

FAKULTAS ILMU KOMPUTER UNIVERSITAS INDONESIA

Algebraic Expression

- ▶ Expression
 - ▶ The fragment of code that calculate new data values
- ▶ The Python interactive shell can be used to evaluate algebraic expressions
 - ▶ $+$, $-$, $/$, $*$
 - ▶ $//$, $\%$, $**$
- ▶ Function
 - ▶ `abs`, `min`, `max`,

Boolean Expression

- ▶ Evaluate Boolean expression (**true** or **false**)
 - ▶ Boolean expressions often involve comparison operators $<$, $>$, $==$, $!=$, $<=$, and $>=$
 - ▶ Boolean expressions may include Boolean operators and, or, and not
- ▶ In a an expression containing algebraic and comparison operators:
 - ▶ Algebraic operators are evaluated first
 - ▶ Comparison operators are evaluated next
 - ▶ Boolean operators are evaluated last
 - ▶ Ex : $2+4 == 2*(9/3)$

Boolean Expression

```
>>> 2<3 and 3<4
True
>>> True and True
True
>>> 4==5 or 3<4
True
>>> False or False
False
>>> not (3<4)
False
>>> 4+1==5 or 4-1<4
True
```

Variables

- ▶ Name we designate to represent an object (number, data structure, function, etc) in our program
- ▶ It is used to make our program more readable and easily understood

▶ Ex:

```
>>>Number = 5
```

```
5
```

```
>>> text = 'This is a text'
```

```
This is a text
```

Reserved Word

- ▶ Not available for you to use as the name of variable, etc in your program

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Variables naming rules

- ▶ (Variable) names can contain these characters:
 - ▶ a through z
 - ▶ A through Z
 - ▶ the underscore character _
 - ▶ digits 0 through 9 (cannot start with digit)
- ▶ Case sensitive
- ▶ For a multiple-word name, use
 - ▶ either the underscore as the delimiter
 - ▶ or camelCase capitalization

Variable and Assignment

- ▶ The assignment statement has the format
 - ▶ **<variable> = <expression>**
 - ▶ **<expression>** is evaluated first and the resulting value is assigned to the **<variable>**

▶ Ex:

```
>>>Number = 5
```

```
5
```

```
>>> result = 5*Number
```

```
25
```


Augmented Assignment

- ▶ Augmented assignment
 - ▶ Combines an operation and assignment
 - ▶ Useful for increment / decrement

shortcut	equivalence
<code>i +=2</code>	<code>i = i+2</code>
<code>i -=2</code>	<code>i = i-2</code>
<code>i /=2</code>	<code>i = i/2</code>
<code>i *=2</code>	<code>i = i*2</code>

Simultaneous Assignment

- ▶ Several values can be assigned at the same time

- ▶ $\langle \text{var}_1 \rangle, \dots, \langle \text{var}_n \rangle = \langle \text{expr}_1 \rangle, \dots, \langle \text{expr}_n \rangle$

- ▶ Ex:

```
>>>x , y = 1 , 8
```

```
>>> x
```

```
1
```

```
>>> y
```

```
8
```

Simultaneous Assignment



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

► Ex:

```
>>>x , y = 1*3 , 8/2
```

```
>>> print (x, y)
```

```
3 4
```

```
>>> x, y = y, x
```

```
>>> print (x, y)
```

```
4 3
```

Variables and types

- ▶ Python doesn't require you to pre-define what type can be associated with a variable
 - ▶ What type variable hold can change
- ▶ Knowing the type can be important for using correct operation
- ▶ The type of variable depends on the type of the object referred to by the variable

Python types

- ▶ Integers (int) : 6
- ▶ Floats (float) : 3.6
- ▶ Booleans (bool) : False
- ▶ String (str) : 'anything', 'Text'
- ▶ List (list) : [], ['2', 4, 6, 'this is text']
- ▶ Others we will learn later

Python types

```
>>> x = 5
```

```
5
```

```
>>> type (x)
```

```
<class, int>
```

```
>>> type (2.5)
```

```
<class, float>
```

```
>>> type ('word')
```

```
<class, str>
```

What is type

- ▶ Type is essentially define two things:
 - ▶ Internal structure of the type (what is contains)
 - ▶ The kind of operations that you can perform
- ▶ `'word'.capitalize ()` is a method that you can call on strings, not in integers
- ▶ Some types have multiple elements (collections), we'll see those later

Converting type

- ▶ A character '1' is not an integer 1
- ▶ You need to convert value returned by the `input` command (a string) into an integer

▶ Ex:

```
>>> variable_a = input ('How old are you?')
```

```
>>> variable_a
```

```
17
```

```
>>> age = int (variable_a)
```

```
17
```


Converting type

- ▶ `int (some_var)` return integer
- ▶ `float (some_var)` return float
- ▶ `str(some_var)` return string
- ▶ Should check out what work:
 - ▶ Ex:
 - ▶ `int (2.1) → 2, int ('2') → 2, but int ('2.1') fails`
 - ▶ `float (2) → 2.0, float ('2.0') → 2.0, float('2') → 2.0, float (2.0) → 2.0`
 - ▶ `str (2) → '2', str (2.0) → '2.0', str('a') → 'a'`

Ariane 5

- ▶ Exploded 37 seconds after liftoff
- ▶ Cargo worth \$500 million
- ▶ Why?
 - ▶ Computed horizontal velocity as floating point number
 - ▶ Converted to 16-bit integer
 - ▶ Worked OK for Ariane 4
 - ▶ Overflowed for Ariane 5
 - ▶ Used same software



Modules

► Modules

- File that can be imported into our python program
- Use other well proven code with yours
 - Math module
 - We `import` a module to use its contents
 - We use the module name as part of the content that we imported
- Ex:

```
>>> import math
>>> print(math.pi)
3.141592653589793
```

Strings

- ▶ A string value is represented as a sequence of characters enclosed within quotes



```
'Hello, World!'
```

- ▶ String values can be manipulated using string operators and functions
- ▶ Ex:

```
>>> 'Hello World'
```

```
Hello World
```

```
>>> s = 'rock'
```

```
>>> t = 'climbing'
```

String Operators



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

Usage	Explanation
<code>x in s</code>	<code>x</code> is a substring of <code>s</code>
<code>x not in s</code>	<code>x</code> is not a substring of <code>s</code>
<code>s + t</code>	Concatenation of <code>s</code> and <code>t</code>
<code>s * n, n * s</code>	Concatenation of <code>n</code> copies of <code>s</code>
<code>s[i]</code>	Character at index <code>i</code> of <code>s</code>
<code>len(s)</code>	(function) Length of string <code>s</code>

```
>>> s = 'rock'
>>> t = 'climbing'
>>> s == 'rock'
True
>>> s != t
True
>>> s < t
False
>>> s > t
True
>>> s + ' ' + t
'rock climbing'
>>> 5 * s
'rockrockrockrockrock'
>>> 30 * '_'
'_____'
>>> 'o' in t
False
>>> 'bi' in t
True
>>> len(t)
8
```

Strings index and indexing operator



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

- ▶ The index of an item in a sequence is its position with respect to the first item
 - ▶ The first item has index 0,
 - ▶ The second has index 1,
 - ▶ The third has index 2, ...

s	=	' A p p l e '
		0 1 2 3 4
s[0]	=	'A'
s[1]	=	'p'
s[2]	=	'p'
s[3]	=	'l'
s[4]	=	'e'

```
>>> s = 'Apple'
>>> s[0]
'A'
>>> s[1]
'p'
>>> s[4]
'e'
```

Strings negative index



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

- ▶ A negative index is used to specify a position with respect to the “end”
 - ▶ The last item has index -1,
 - ▶ The second to last item has index -2,
 - ▶ The third to last item has index -3, ...

		-5	-4	-3	-2	-1
s	=	' A p p l e '				
		0	1	2	3	4
s[-1]	=					'e'
s[-2]	=				'l'	
s[-5]	=	'A'				

```
>>> s = 'Apple'
>>> s[-1]
'e'
>>> s[-2]
'l'
>>> s[-5]
'A'
```


List



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

- ▶ Coma-separated sequence of items enclosed within square bracket



```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

- ▶ The items can be numbers, strings, and even other lists

```
>>> pets = ['ant', 'bat', 'cod', 'dog', 'elk']  
>>> lst = [0, 1, 'two', 'three', [4, 'five']]  
>>> nums = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```


List Operators and Function



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

Usage	Explanation
<code>x in lst</code>	<code>x</code> is an item of <code>lst</code>
<code>x not in lst</code>	<code>x</code> is not an item of <code>lst</code>
<code>lst + lstB</code>	Concatenation of <code>lst</code> and <code>lstB</code>
<code>lst*n, n*lst</code>	Concatenation of <code>n</code> copies of <code>lst</code>
<code>lst[i]</code>	Item at index <code>i</code> of <code>lst</code>
<code>len(lst)</code>	Number of items in <code>lst</code>
<code>min(lst)</code>	Minimum item in <code>lst</code>
<code>max(lst)</code>	Maximum item in <code>lst</code>
<code>sum(lst)</code>	Sum of items in <code>lst</code>

```
>>> lst = [1, 2, 3]
>>> lstB = [0, 4]
>>> 4 in lst
False
>>> 4 not in lst
True
>>> lst + lstB
[1, 2, 3, 0, 4]
>>> 2*lst
[1, 2, 3, 1, 2, 3]
>>> lst[0]
1
>>> lst[1]
2
>>> lst[-1]
3
>>> min(lst)
1
>>> max(lst)
3
>>> sum(lst)
6
```



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

List are mutable, strings are not

- List can be modified (mutable)

```
pets = ['ant', 'bat', 'cow', 'dog', 'elk']
```

- String cannot be modified (immutable)

```
pet = 'cod'
```

```
>>> pets = ['ant', 'bat', 'cod', 'dog', 'elk']
```

```
>>> pets[2] = 'cow'
```

```
>>> pets
```

```
['ant', 'bat', 'cow', 'dog', 'elk']
```

```
>>> pet = 'cod'
```

```
>>> pet[2] = 'w'
```

```
Traceback (most recent call last):
```

```
  File "<pyshell#155>", line 1, in <module>
```

```
    pet[2] = 'w'
```

```
TypeError: 'str' object does not support item assignment
```

List Method

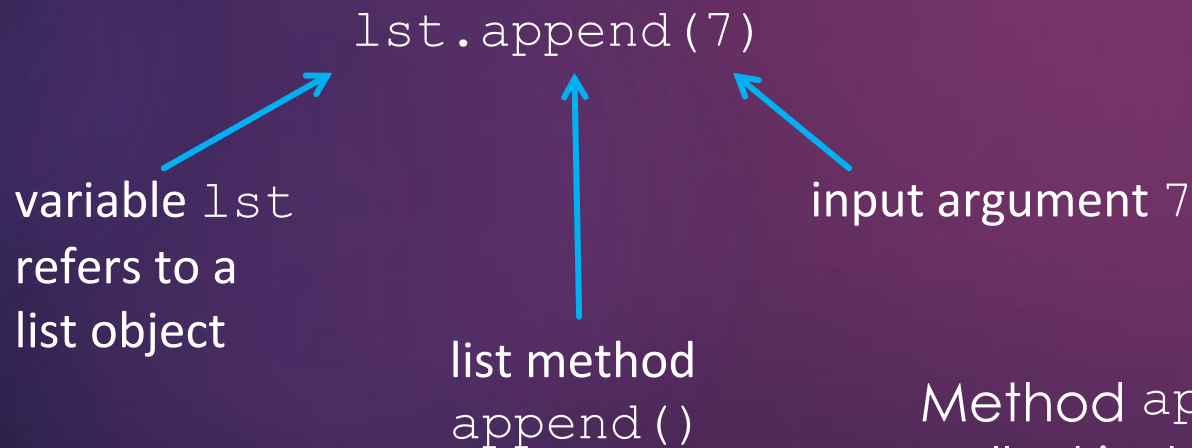


UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

- ▶ `len()` and `sum()` are examples of functions that can be called with the list input argument

```
>>> lst = [1, 2, 3]
>>> len(lst)
3
>>> sum(lst)
6
>>> lst.append(7)
>>> lst
[1, 2, 3, 7]
>>>
```



Method `append()` can't be called independently; it must be called on some list object

List methods



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

Usage	Explanation
<code>lst.append(item)</code>	adds <code>item</code> to the end of <code>lst</code>
<code>lst.count(item)</code>	returns the number of times <code>item</code> occurs in <code>lst</code>
<code>lst.index(item)</code>	Returns index of (first occurrence of) <code>item</code> in <code>lst</code>
<code>lst.pop()</code>	Removes and returns the last item in <code>lst</code>
<code>lst.remove(item)</code>	Removes (the first occurrence of) <code>item</code> from <code>lst</code>
<code>lst.reverse()</code>	Reverses the order of items in <code>lst</code>
<code>lst.sort()</code>	Sorts the items of <code>lst</code> in increasing order

Methods `append()`, `remove()`, `reverse()`, and `sort()` do not return any value; they, along with method `pop()`, modify list `lst`

```
>>> lst = [1, 2, 3]
>>> lstB = [0, 4]
>>> 4 in lst
False
>>> 4 not in lst
True
>>> lst + lstB
[1, 2, 3, 0, 4]
>>> 2*lst
[1, 2, 3, 1, 2, 3]
>>> lst[0]
1
>>> lst[1]
2
>>> lst[-1]
3
>>> min(lst)
1
>>> max(lst)
3
>>> sum(lst)
6
```

Built in Class Tuple



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

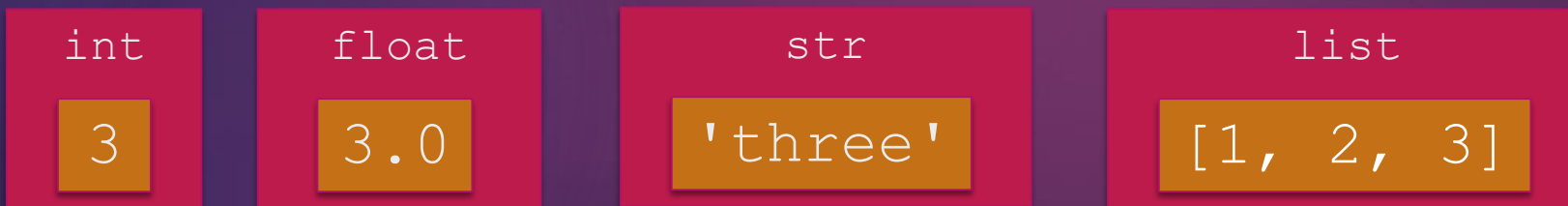
FACULTY OF
COMPUTER
SCIENCE

- ▶ The class tuple is same as list except that tuple is **immutable**
- ▶ Sometimes we need to have an “immutable list”

```
>>> lst = ['one', 'two', 3]
>>> lst[2] = 'three'
>>> lst
['one', 'two', 'three']
>>> tpl = ('one', 'two', 3)
>>> tpl[2]
3
>>> tpl[2] = 'three'
Traceback (most recent call last):
  File "<pyshell#131>", line 1, in
<module>
    tpl[2] = 'three'
TypeError: 'tuple' object does not
support item assignment
```

Object and Classes

- ▶ In python, every value is stored as object in memory
- ▶ Every object has value and type
- ▶ It is the object that has a type, not the variable



An object's type determines what values it can have and how it can be manipulated

```
>>> a = 3
>>> b = 3.0
>>> c = 'three'
>>> d = [1, 2, 3]
>>> type(a)
<class 'int'>
>>> type(b)
<class 'float'>
>>> type(c)
<class 'str'>
>>> type(d)
<class 'list'>
>>> a = []
>>> type(a)
<class 'list'>
```

Terminology: object X is of type `int` = object X belongs to class `int`

Operator for Number Types



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

- ▶ An object's type determines what values it can have and how it can be manipulated
- ▶ We already saw the operators that are used to manipulate number types
 - ▶ algebraic operators `+`, `-`, `*`, `/`, `//`, `%`, `**`, `abs()`
 - ▶ comparison operators `>`, `<`, `==`, `!=`, `<=`, `>=`, ...
- ▶ Parentheses and precedence rules determine the order in which operators are evaluated in an expression

higher
precedence

Operator

[...]

`x[]`

`**`

`+x`, `-x`

`*`, `/`, `//`, `%`

`+`, `-`

`in`, `not in`

`<`, `>`, `<=`, `>=`, `==`, `!=`

`not x`

`and`

`or`

lower
precedence

Object Constructor



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

- ▶ Implicitly assigning object type
 - ▶ Integer object can be defined by assigning with value 3
- ▶ Explicitly assigning object type
 - ▶ `int ()` : integer constructor (default value: 0)
 - ▶ `float ()` : float constructor (default value: 0.0)
 - ▶ `str ()` : string constructor (default value: '')
 - ▶ `bool ()` : Boolean constructor (default value: False)
 - ▶ `list ()` : List constructor (default value : [])

```
>>> x = 3
>>> x
3
>>> x = int(3)
>>> x
3
>>> x = int()
>>> x
0
>>> y = float()
>>> y
0.0
>>> s = str()
>>> s
''
>>> lst = list()
>>> lst
[]
```


Type Conversion

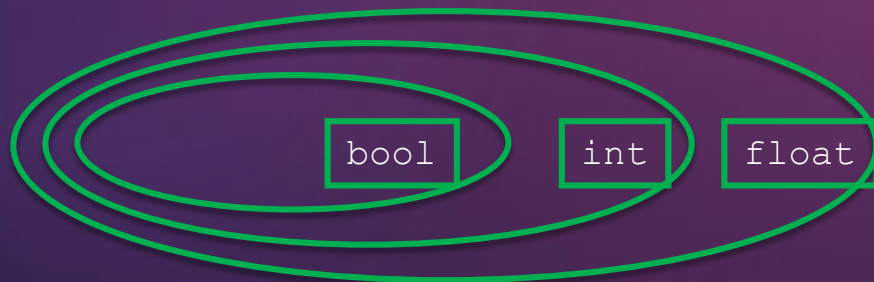


UNIVERSITAS
INDONESIA
Veritas, Probitas, Justitia

FACULTY OF
COMPUTER
SCIENCE

► Implicit type conversion

- When evaluating an expression that contains operands of different type, operands must first be converted to the same type
- Operands are converted to the type that “contains the others”



```
>>> x = 3
>>> x+2.0
5.0
>>> False + 2
2
>>> True+True
2
>>> x = 3.0
>>> x+4
7.0
```

Python Standard Library

- ▶ The core Python programming language comes with functions such as `max()` and `sum()` and classes such as `int`, `str`, and `list`.
- ▶ Python Standard Library to support
 - ▶ Network programming
 - ▶ Web application programming
 - ▶ Graphical user interface (GUI) development
 - ▶ Database programming
 - ▶ Mathematical functions
 - ▶ Pseudorandom number generators
 - ▶ Media processing, etc.
- ▶ The Python Standard Library functions and classes are organized into components called **modules**.