

IT5001 Software Development Fundamentals

8. Anonymous Functions
Sirigina Rajendra Prasad

Functions

- Traditional Form
 - $\text{square}(x) \mapsto x^2$
- Anonymous Form:
 - $x \mapsto x^2$

Anonymous Functions

- Also known as lambda (λ) functions
 - Functions without name
- Python Syntax:
 - `lambda args: expression`

Functions: Traditional Maths vs λ -Calculus

Mathematics

- Abstraction
 - *square*: $x \mapsto x^2$
 - *square*(x) = x^2
- Application
 - *square*(5) = $5^2 = 25$

λ -Calculus

- Abstraction
 - $x \mapsto x^2$
 - $\lambda x. x^2$
- Application
 - $(\lambda x. x^2)5 = 5^2 = 25$

Functions: λ -Calculus vs Python

λ -Calculus

- Abstraction
 - $x \mapsto x^2$
 - $\lambda x. x^2$
- Application
 - $(\lambda x. x^2)5 = 5^2 = 25$

Python

- Abstraction

```
>>> lambda x: x**2
<function <lambda> at 0x000002557882E708>
```
- Application

```
>>> (lambda x: x**2) (5)
25
```

or

```
>>> f = lambda x: x**2+1
```

```
>>> f(5)
```

```
26
```

Application

Abstraction

Examples: Identity Function

λ -Calculus

- λ -Calculus
 - $(\lambda x. x)$
 - Takes a variable x as input argument
 - returns x

Python

```
>>> lambda x:x
<function <lambda> at 0x000002557882E3A8>
>>> (lambda x:x)('abc')
'abc'
>>> (lambda x:x)(10)
10
```

Examples: Constant Function

λ -Calculus

- λ -Calculus
 - $(\lambda x. y)$
 - Takes a variable x and return y

Python

```
>>> (lambda x: 'abc') (5)
'abc'
```

```
>>> (lambda x: 'abc') (10)
'abc'
```

Alternative in Python:

```
>>> (lambda : 10) ()
10
>>> x = lambda : 10
>>> x()
10
```

Anonymous Functions: More Examples

- Can pass multiple arguments as inputs

```
>>> lambda x,y,z: x+y+z
<function <lambda> at 0x0000021EAA3B9DC8>
>>> (lambda x,y,z: x+y+z) (4,5,9)
18
```

```
>>> my_list = [1,2,3]
>>> (lambda x,y,z: x+y+z) (my_list)
Traceback (most recent call last):
  File "<pyshell#27>", line 1, in <module>
    (lambda x,y,z: x+y+z) (my_list)
TypeError: <lambda>() missing 2 required positional arguments: 'y' and 'z'
>>> (lambda my_list: my_list[0]+my_list[1]+my_list[2]) (my_list)
6
```