

Welcome



:Today's Topic: BLOCK CHAIN

:Simple Block Chain Through Python Code:

:Today's Overview:[Python Part]



REPL vs Files + IDE

Variables & Data Types

Operators

Lists

Functions

:Today's Overview:[Block Chain Part]



Manage a List of Blocks



List of Data ("Coins")



User adds ("mines") Data



Output Blockchain

REPL

Read

2 + 2

Evaluate

...

Print

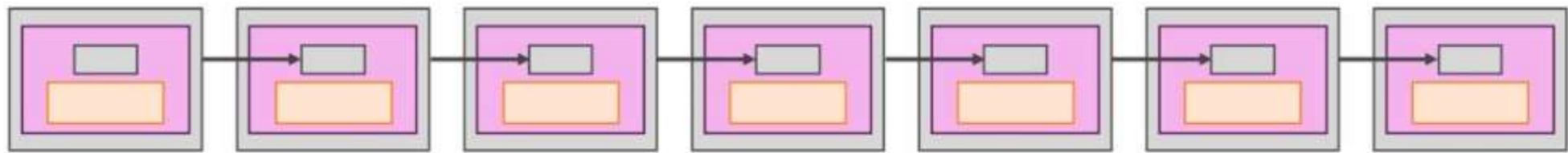
4

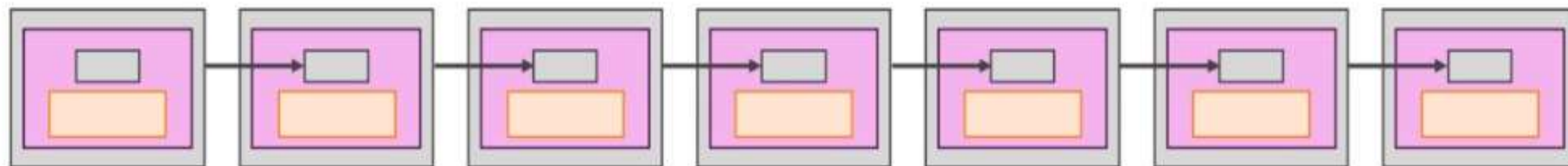
Loop



Interactive Python Programming, great for simple calculations and practicing

A Simple Representation





The Data



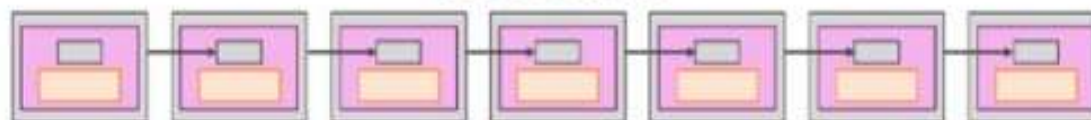
A Number

A String

A Boolean

Complex Structure

The Chain



A List

Data Types

Numbers

Integer

10

-3

Floats

1.8

-6.973

Booleans

True

False

Strings

'Hi there!'

"Hi there!"

Complex Types like Dictionaries & Objects

Numbers

Integer

10, -55, 5421

As big (small) as supported by your
Memory and Operating System

Convert other types to Integer with
`int()`

Float

1.591, -0.81, 5000.0

As big (small) as supported by your
Memory and Operating System

Convert other types to Float with
`float()`

Write long numbers in easily readable way: `1_000_000.0`

Operators

+

5 + 10

Works with Strings

15

-

5 - 10

-5

*

5 * 10

Works with Strings

50

/

5 / 10

Float

0.5

//

5 // 10

0

**

5 ** 10

9765625

%

5 % 10

5

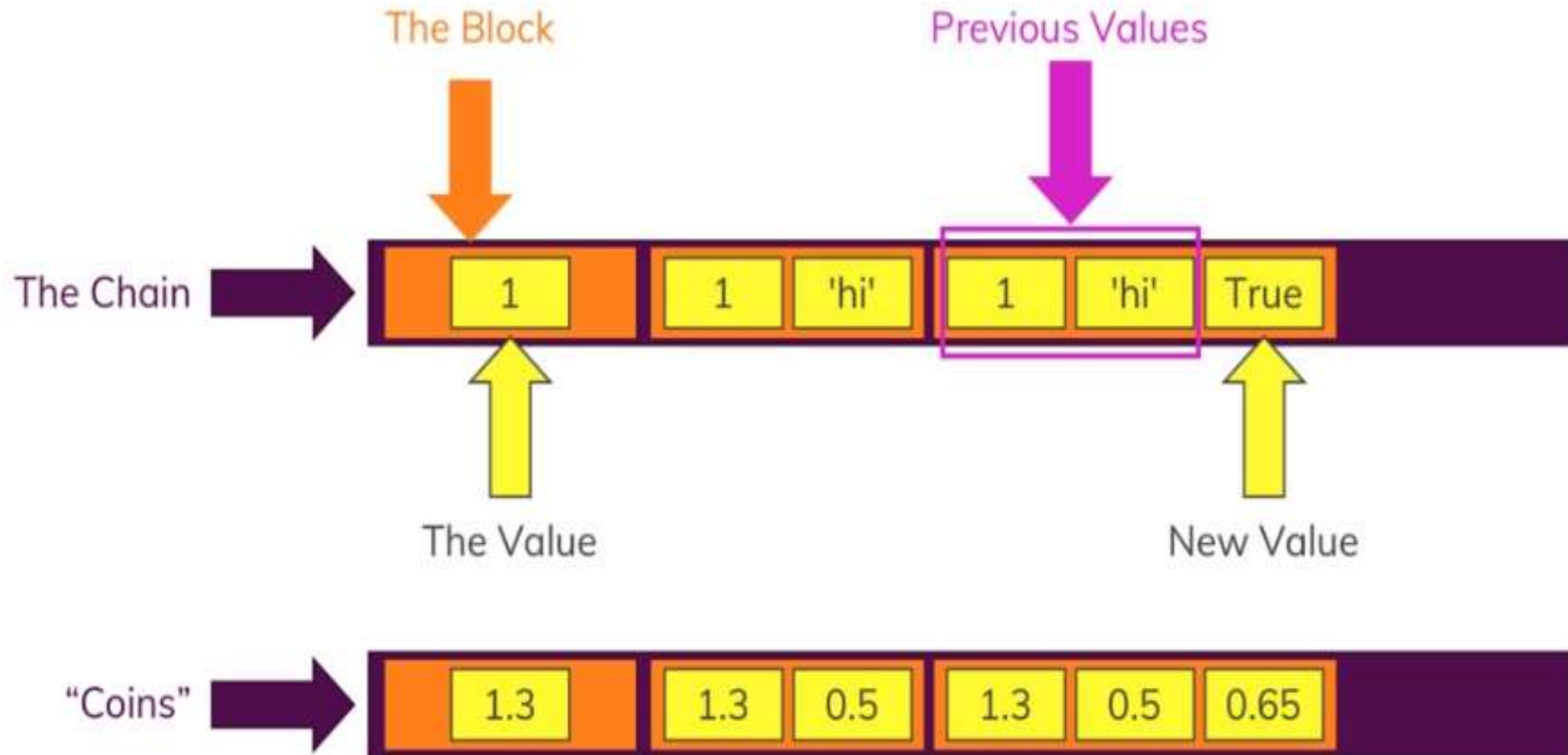
List

Index	0	1	2	3
	"some text",	12.9 ,	True ,	["nested!", 8]

Length = 4

my_list[0] yields "some text"

Block Chain[Theory]



Functions

Define Code which is executed Later (and possibly Multiple Times)

```
def greet():  
    print('Hello')  
  
greet()  
greet()
```

Functions [with args]

Define Code which is executed Later (and possibly Multiple Times)

```
def greet():  
    print('Hello')  
  
greet()  
greet()
```

Can receive Arguments

```
def greet(name):  
    print('Hello ' + name)  
  
greet('Max')
```


Variable Scope

Global

```
name = 'Max'  
def greet():  
    print('Hi , + name)  
  
greet()
```

Local

```
name = 'Max'  
def greet():  
    age = 29  
    print('Hi , + name + ', I am' + age)  
  
greet()
```

:Summery:

Indentation structures Code

Block statements need a **:**

Functions defined with **def**

Follow **PEB 8** Code Style

Data Types

- Numbers (Integers & Floats)
- Strings
- Booleans

Lists

- Create Lists
- Add Items via `append()`
- Access Items **via Index** (which **starts at 0!**)
- Other List Operations (`pop()`, ...)

Operators

- Base Arithmetic: `+`, `-`, `*`, `/`
- Modulus: `%` (`15 % 10 = 5`)
- Floor Division: `//` (`15 // 10 = 1`)
- Power: `**` (`2 ** 3 = 8`)
- Strings can be **added** and **multiplied** (with Integers)

Scope

- **Global:** Variables defined outside of Functions
- **Local:** Variables defined inside of Functions

Functions

- Use **Indentation** and **:** to define Code Block
- Can use **Arguments**
- Can **return** Values
- Can use **Default Arguments**
- **Keyword Arguments** allow you to re-order or skip arguments

Our Status

- | | | |
|--|---|----------------------|
| <input checked="" type="checkbox"/> Chain of Data | ➡ | Basic Implementation |
| <input checked="" type="checkbox"/> Mine new Blocks | ➡ | Basic Implementation |
| <input checked="" type="checkbox"/> Block Hashing | ➡ | Basic Implementation |
| <input type="checkbox"/> Analyze & Verify Chain | | |
| <input type="checkbox"/> Real Transactions | | |
| <input type="checkbox"/> Store Chain to Disk | | |
| <input type="checkbox"/> Node Network | | |
| <input type="checkbox"/> Share Data, Resolve Conflicts | | |
| <input type="checkbox"/> Wallets | | |

Thank
you

