encode CLUB

Algorand Bootcamp

Observations

- TxnType.ApplicationCall **IS NOT** Txn.on_competion()

- Application Array can take upto 8 Applications, not 16*

- Btoi() must be used to convert argument from byte to int. You cannot typecast it with Int()

- Argument must be an integer before performing mathematical operations on them

Observations

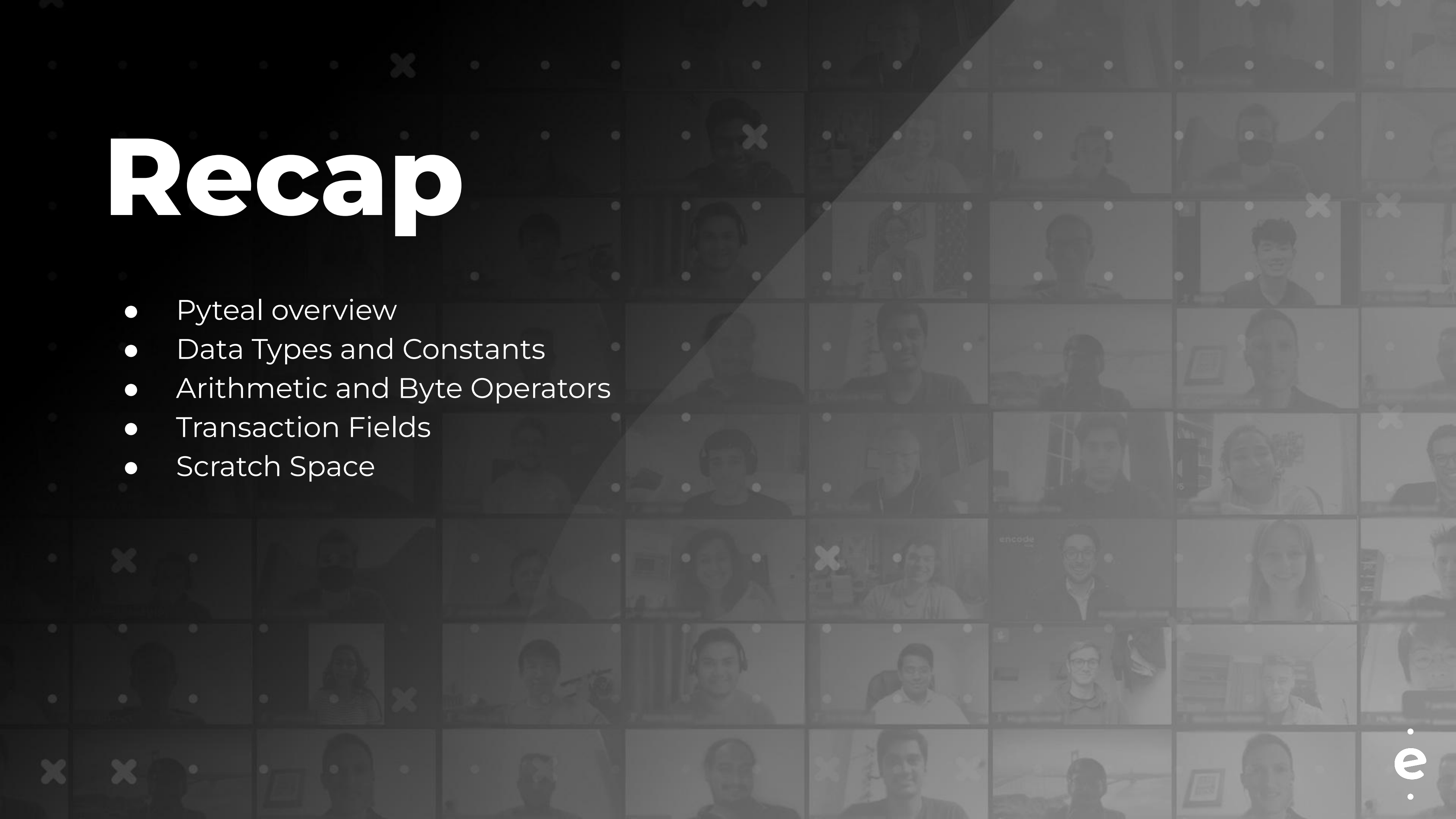
- To validate asset ID in PyTeal, you must use Txn.xfer_asset()

- You CANNOT stop an account from ever clearing it’s local state. You can perform any final changes before it tries to clear state, but you cannot stop it from happening.

- More on smart contract arrays: https://developer.algorand.org/docs/get-details/dapps/smart-contracts/apps/?from_query=applications%20array#smart-contract-arrays

Any questions from previous session?

Any issues with the assignment?

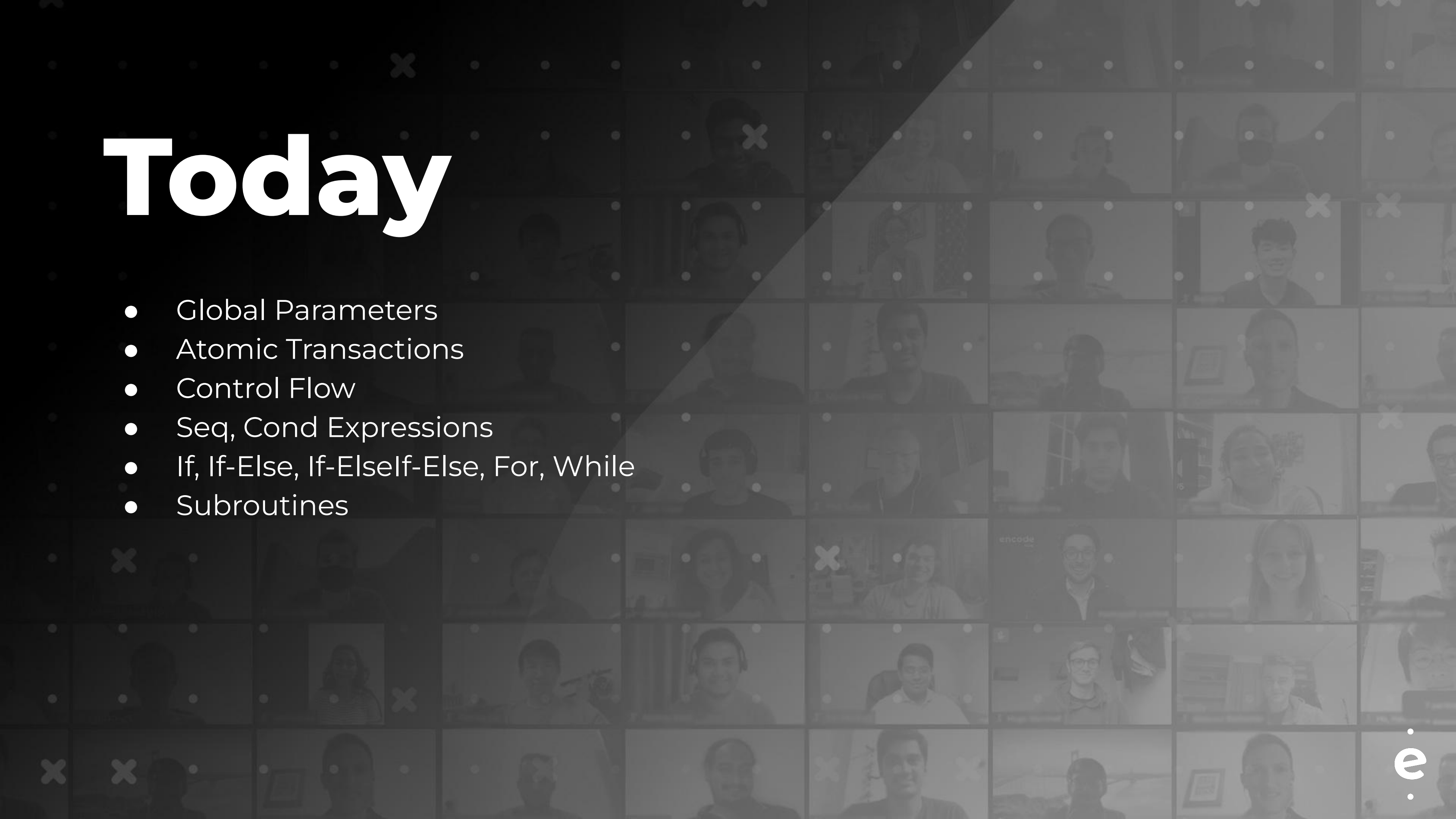


Recap

- Pyteal overview
- Data Types and Constants
- Arithmetic and Byte Operators
- Transaction Fields
- Scratch Space

Today

- Global Parameters
- Atomic Transactions
- Control Flow
- Seq, Cond Expressions
- If, If-Else, If-ElseIf-Else, For, While
- Subroutines



Global Parameters

- On-chain parameters available within PyTeal contracts

- Useful in assertions and logic validations

Global Parameters

- Global.zero_address()
- Global.latest_timestamp()
- Global.current_application_address()
- Global.creator_address()
- Global.current_application_id()

Atomic Transactions

- Group of transactions that either ensure all go through or none go through
- Useful when having relatively complicated transaction logic when order of transactions is important
- Transactions accessed by Gtxn object

Atomic Transactions

- If transactions are grouped, they will have a group ID
- Even if one of the transactions fail, all the other transactions in the group fail
- Use case: Grouping a payment call with an application transaction call

Control Flow

Expressions

- **Approve()**
  - Similar to Return(Int(1)) - Approve the transaction at the current instruction, don’t go any further
- **Reject()**
  - Similar to Return(Int(0) - Reject the transaction at the current instruction, don’t go any further
- **Assert()**
  - The Assert expression can be used to ensure that conditions are met before continuing the program.

Conditionals

Cond([test-expr-1, body-1],

[test-expr-2, body-2], ...)

- Each test-expr is evaluated in order.

- If it produces 0, the paired body is ignored, and evaluation proceeds to the next test-expr.

Conditionals

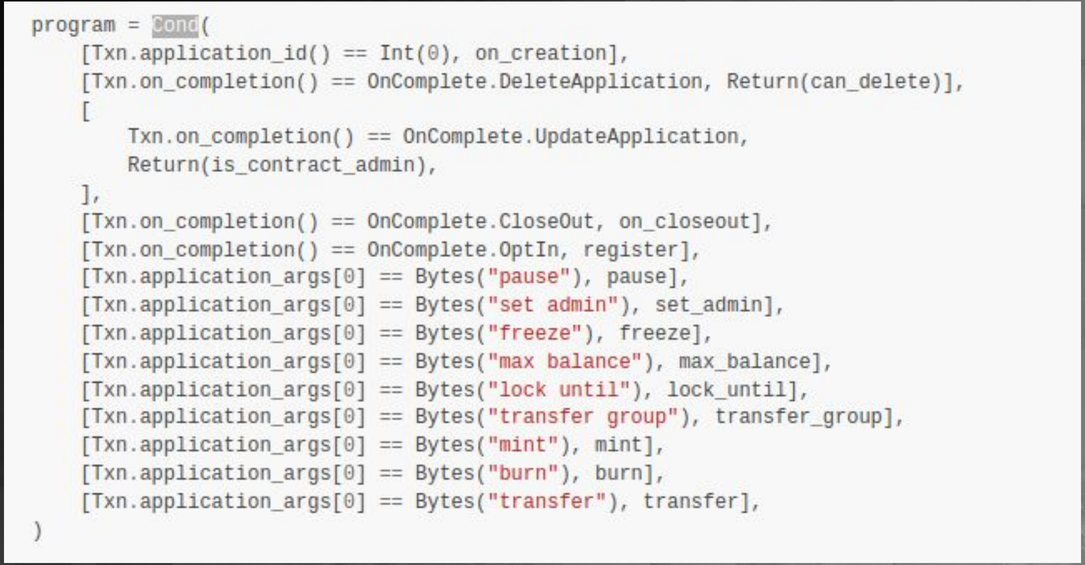
Cond([test-expr-1, body-1],

[test-expr-2, body-2], ……….)

- As soon as a test-expr produces a true value (> 0), its body is evaluated to produce the value for this Cond expression.

- If none of test-expr s evaluates to a true value, the Cond expression will be evaluated to err, a TEAL opcode that causes the runtime panic.

Conditionals



```
program = Cond(
    [Txn.application_id() == Int(0), on_creation],
    [Txn.on_completion() == OnComplete.DeleteApplication, Return(can_delete)],
    [
        Txn.on_completion() == OnComplete.UpdateApplication,
        Return(is_contract_admin),
    ],
    [Txn.on_completion() == OnComplete.CloseOut, on_closeout],
    [Txn.on_completion() == OnComplete.OptIn, register],
    [Txn.application_args[0] == Bytes("pause"), pause],
    [Txn.application_args[0] == Bytes("set admin"), set_admin],
    [Txn.application_args[0] == Bytes("freeze"), freeze],
    [Txn.application_args[0] == Bytes("max balance"), max_balance],
    [Txn.application_args[0] == Bytes("lock until"), lock_until],
    [Txn.application_args[0] == Bytes("transfer group"), transfer_group],
    [Txn.application_args[0] == Bytes("mint"), mint],
    [Txn.application_args[0] == Bytes("burn"), burn],
    [Txn.application_args[0] == Bytes("transfer"), transfer],
)
```

Source: https://pyteal.readthedocs.io/en/stable/examples.html

If

If(test-expr-1, then-expr, else-expr)

- the test-expr is always evaluated and needs to be typed TealType.uint64.

- If it results in a value greater than 0, then the then-expr is evaluated.

- Otherwise, else-expr is evaluated.

If

If(test-expr-1, then-expr)

- Can skip *else-expr*

If(test-expr)

.Then(then-expr)

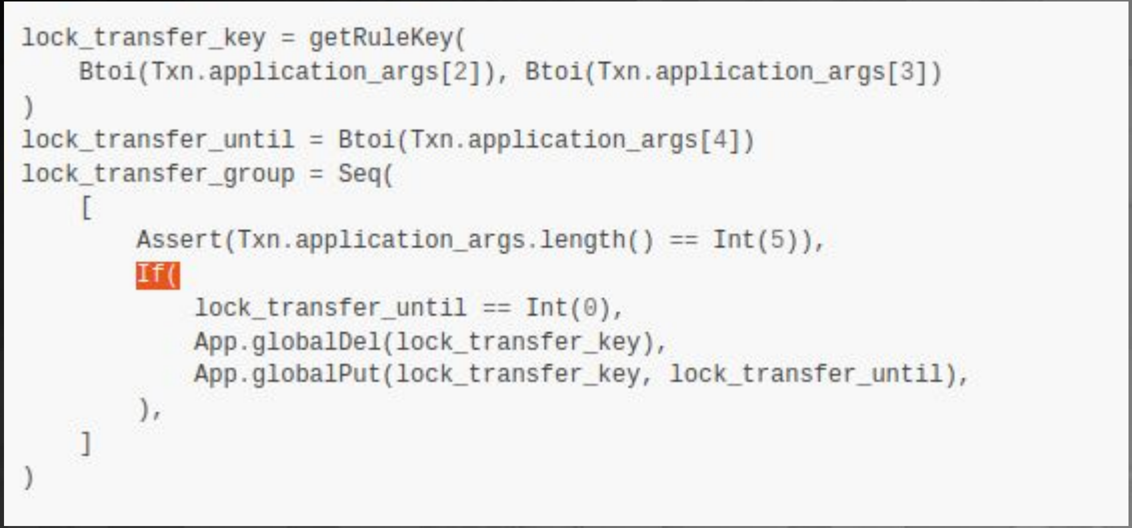
.ElseIf(test-expr)

.Then(then-expr)

.Else(else-expr)

- If-ElseIf-Else chain is also possible

If



```
lock_transfer_key = getRuleKey(
    Btoi(Txn.application_args[2]), Btoi(Txn.application_args[3])
)
lock_transfer_until = Btoi(Txn.application_args[4])
lock_transfer_group = Seq(
    [
        Assert(Txn.application_args.length() == Int(5)),
        If(
            lock_transfer_until == Int(0),
            App.globalDel(lock_transfer_key),
            App.globalPut(lock_transfer_key, lock_transfer_until),
        ),
    ]
)
```

Source: https://pyteal.readthedocs.io/en/stable/examples.html

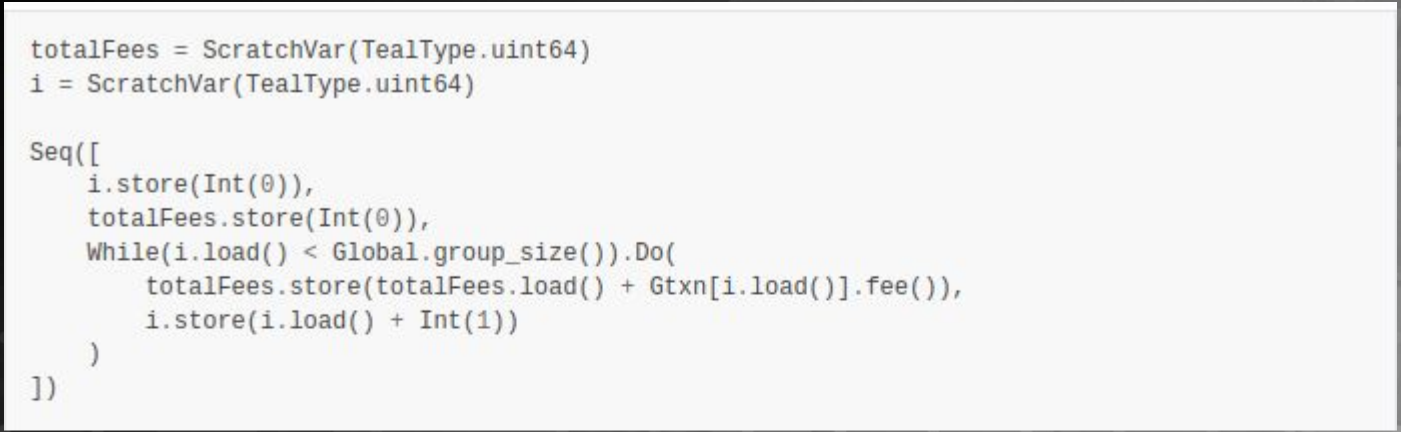
While

While(loop-condition).Do(loop-body)

- The loop-condition expression must evaluate to TealType.uint64, and the loop-body expression must evaluate to TealType.none.

- The loop-body expression will continue to execute as long as loop-condition produces a true value (> 0).

While



```
totalFees = ScratchVar(TealType.uint64)
i = ScratchVar(TealType.uint64)

Seq([
    i.store(Int(0)),
    totalFees.store(Int(0)),
    While(i.load() < Global.group_size()).Do(
        totalFees.store(totalFees.load() + Gtxn[i.load()].fee()),
        i.store(i.load() + Int(1))
    )
])
```

Source: https://pyteal.readthedocs.io/en/stable/control_structures.html

For

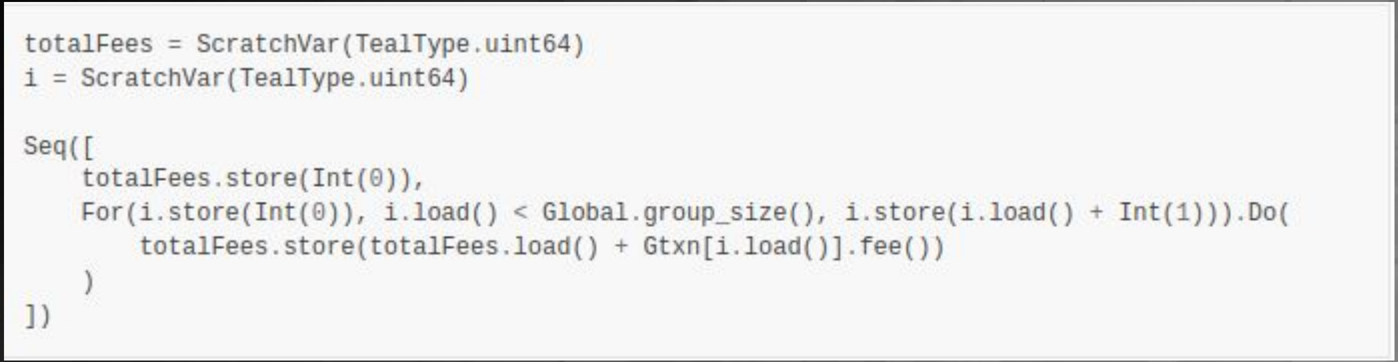
For(loop-start, loop-condition, loop-set).Do(loop-body)

- The loop-start, loop-step, and loop-body expressions must evaluate to TealType.none, and the the loop-condition expression must evaluate to *TealType.uint64*.

- When a For expression is executed, loop-start is executed first.

- Then the expressions loop-condition, loop-body, and loop-step will continue to execute in order as long as loop-condition produces a true value (> 0).

For



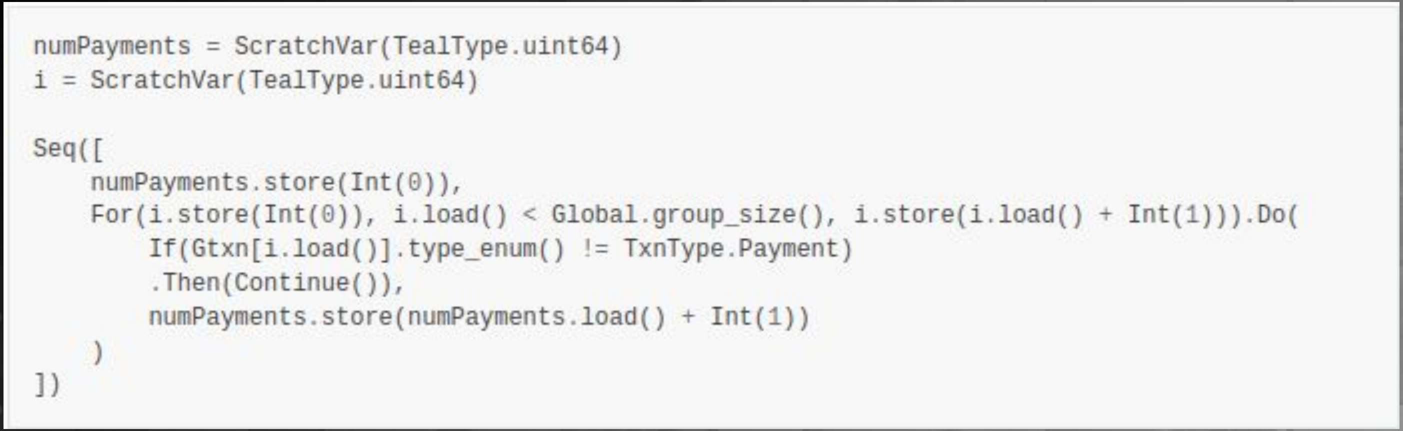
```
totalFees = ScratchVar(TealType.uint64)
i = ScratchVar(TealType.uint64)

Seq([
    totalFees.store(Int(0)),
    For(i.store(Int(0)), i.load() < Global.group_size(), i.store(i.load() + Int(1))).Do(
        totalFees.store(totalFees.load() + Gtxn[i.load()].fee())
    )
])
```

Source: https://pyteal.readthedocs.io/en/stable/control_structures.html

Exiting Loops

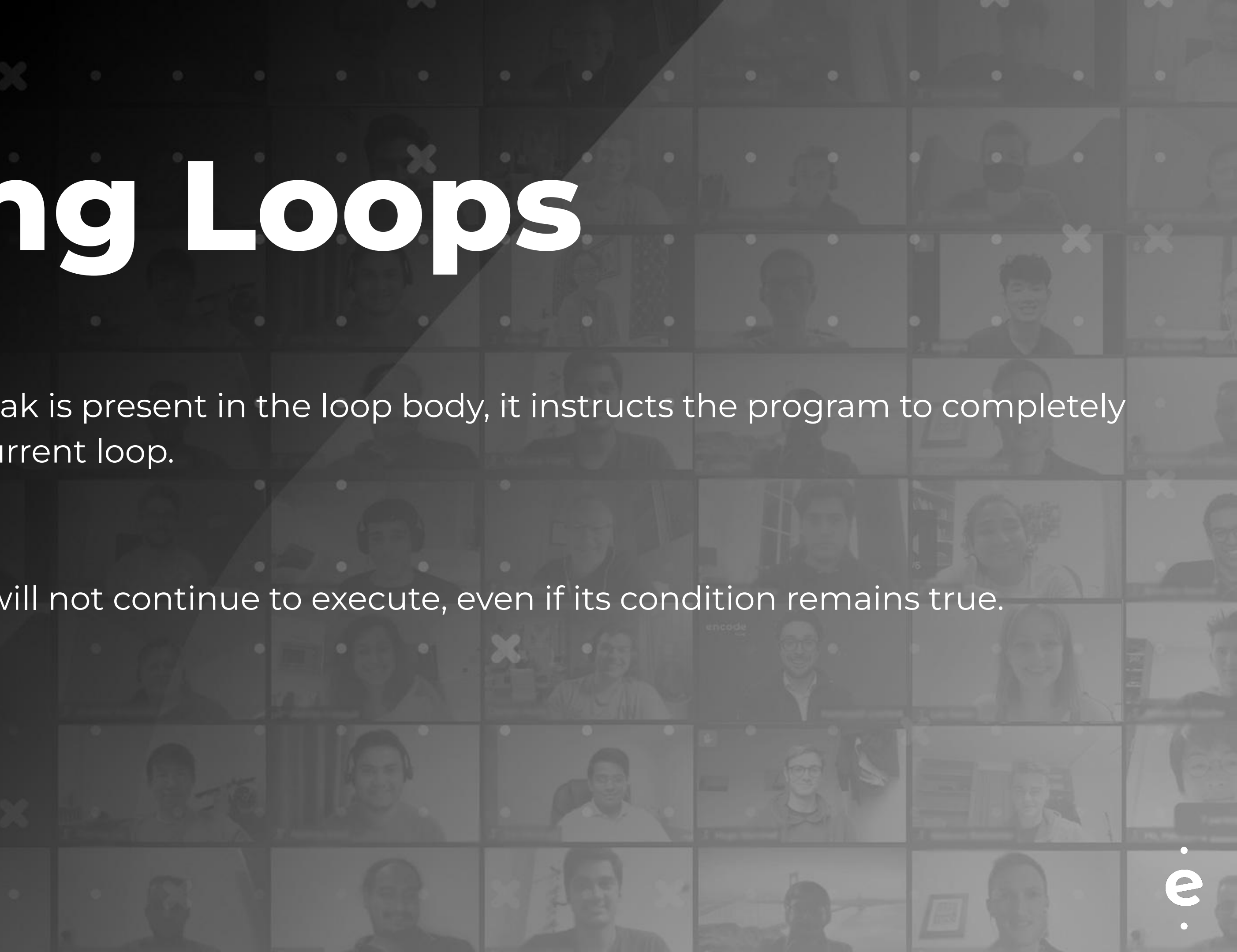
- Continue()
  - When Continue is present in the loop body, it instructs the program to skip the remainder of the loop body.
  - The loop may continue to execute as long as its condition remains true.

Exiting Loops



```
numPayments = ScratchVar(TealType.uint64)
i = ScratchVar(TealType.uint64)

Seq([
    numPayments.store(Int(0)),
    For(i.store(Int(0)), i.load() < Global.group_size(), i.store(i.load() + Int(1))).Do(
        If(Gtxn[i.load()].type_enum() != TxnType.Payment)
        .Then(Continue()),
        numPayments.store(numPayments.load() + Int(1))
    )
])
```

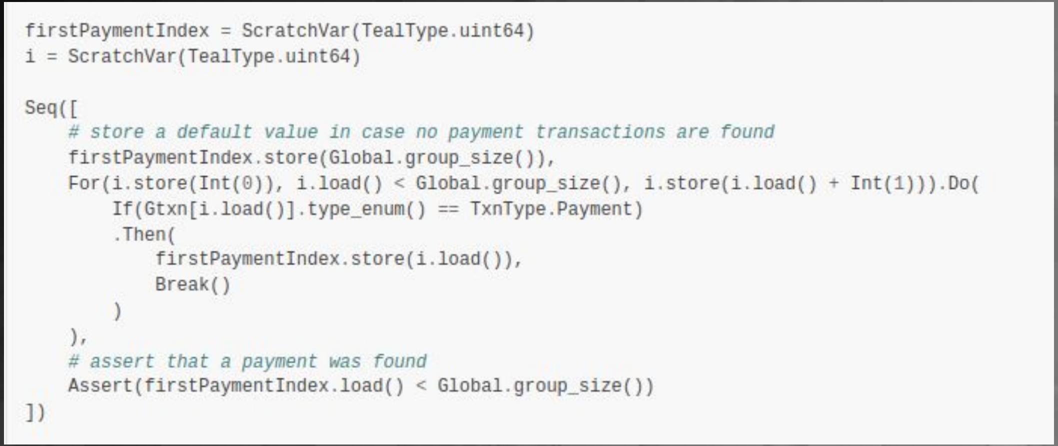
Source: https://pyteal.readthedocs.io/en/stable/control_structures.html



Exiting Loops

- Break()
  - When Break is present in the loop body, it instructs the program to completely exit the current loop.
  - The loop will not continue to execute, even if its condition remains true.

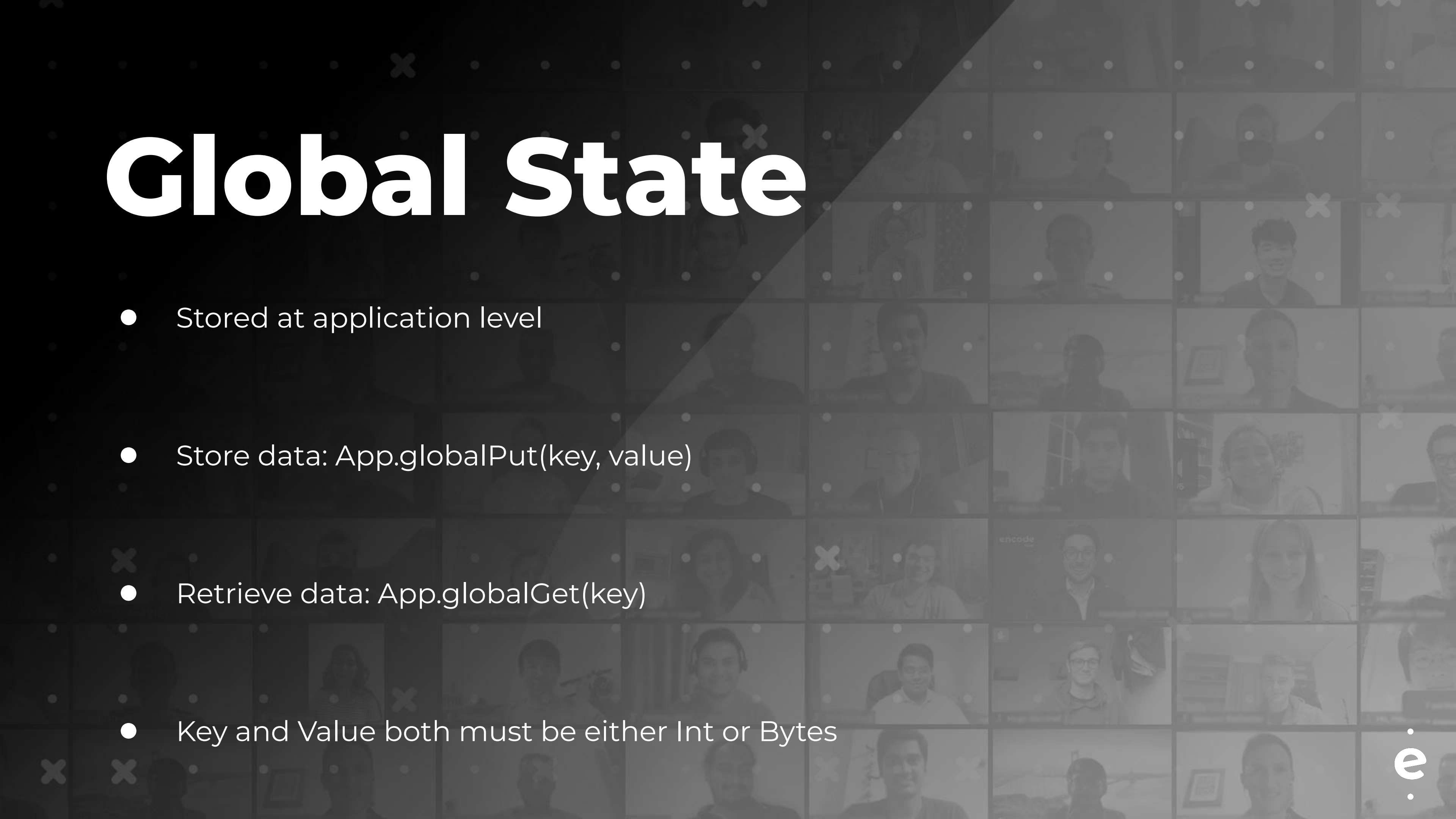
Exiting Loops



```
firstPaymentIndex = ScratchVar(TealType.uint64)
i = ScratchVar(TealType.uint64)

Seq([
    # store a default value in case no payment transactions are found
    firstPaymentIndex.store(Global.group_size()),
    For(i.store(Int(0)), i.load() < Global.group_size(), i.store(i.load() + Int(1))).Do(
        If(Gtxn[i.load()].type_enum() == TxnType.Payment)
        .Then(
            firstPaymentIndex.store(i.load()),
            Break()
        )
    ),
    # assert that a payment was found
    Assert(firstPaymentIndex.load() < Global.group_size())
])
```

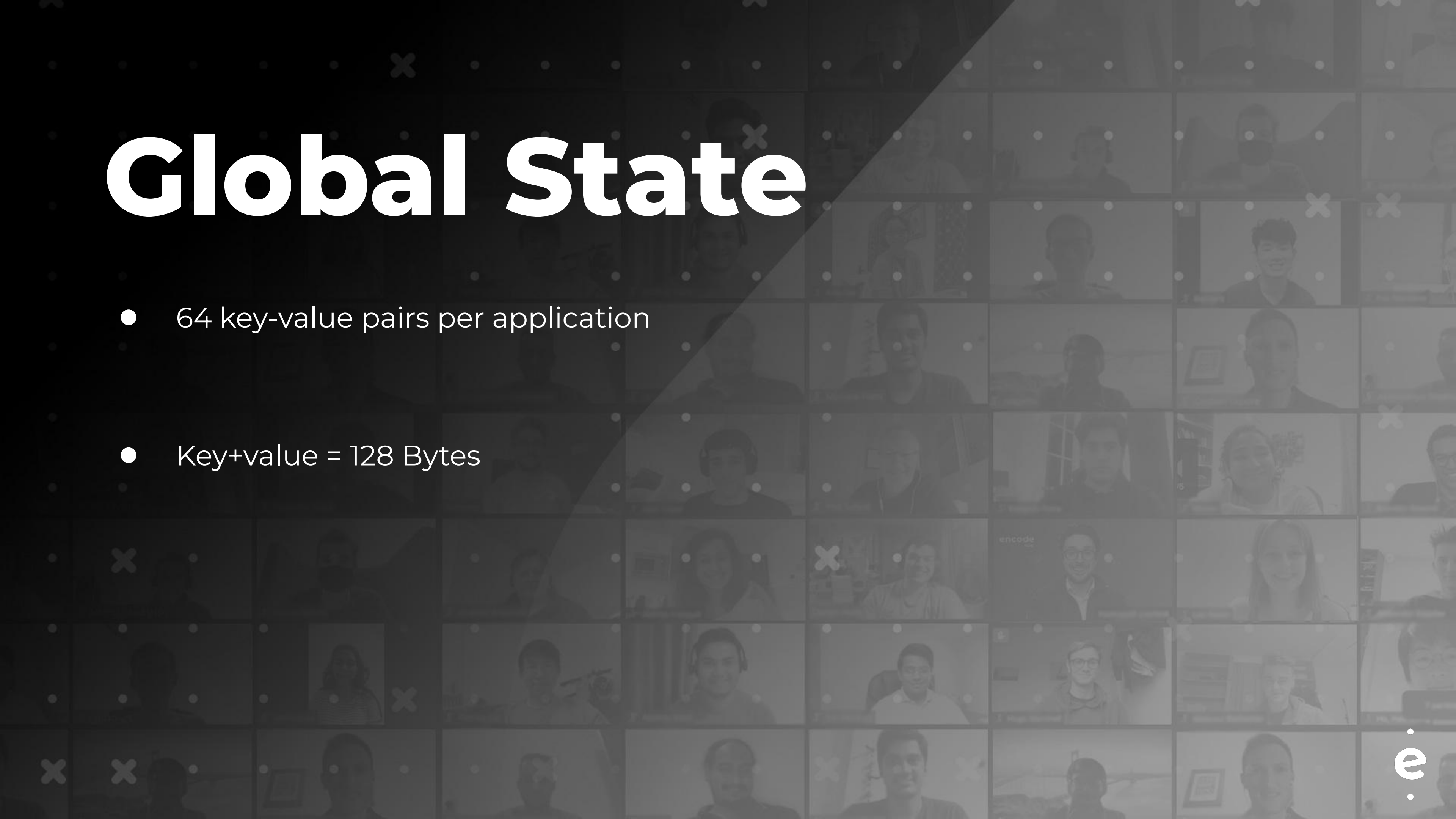
Source: https://pyteal.readthedocs.io/en/stable/control_structures.html

Primer on State Management



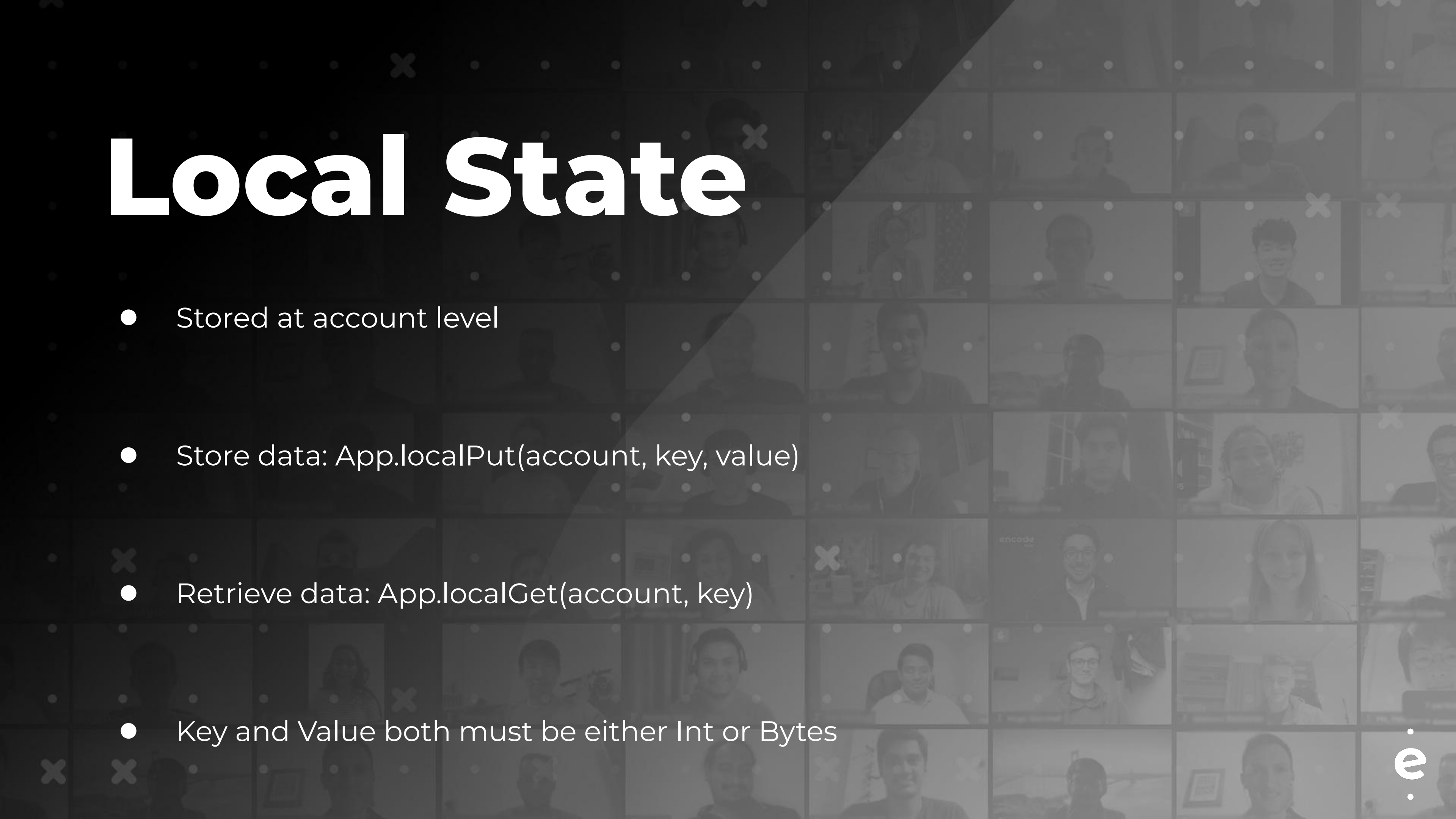
Global State

- Stored at application level
- Store data: App.globalPut(key, value)
- Retrieve data: App.globalGet(key)
- Key and Value both must be either Int or Bytes



Global State

- 64 key-value pairs per application
- Key+value = 128 Bytes



Local State

- Stored at account level
- Store data: App.localPut(account, key, value)
- Retrieve data: App.localGet(account, key)
- Key and Value both must be either Int or Bytes

Enough Theory, Let’s Code

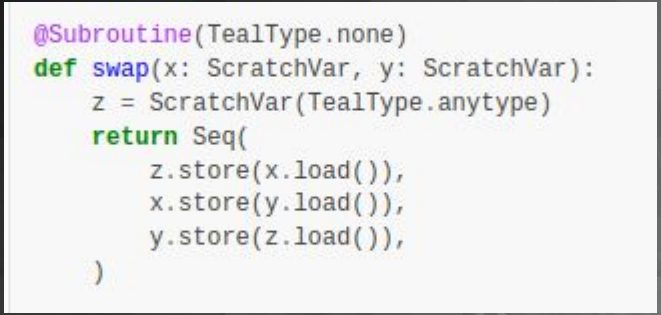
Subroutines

Subroutines

Similar to functions, a subroutine is section of code that can be called multiple times from within a program.

- Subroutines accept any number of arguments.
- Subroutine argument types can be any Expr (PyTeal expression) or strictly ScratchVar (no subclasses allowed).
- Subroutines return a single value, or no value.

Subroutine

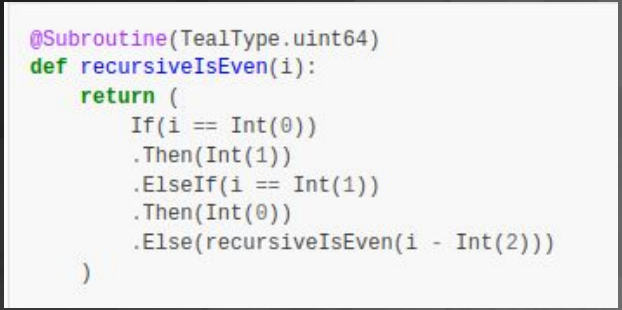


```
@Subroutine(TealType.none)
def swap(x: ScratchVar, y: ScratchVar):
    z = ScratchVar(TealType.anytype)
    return Seq(
        z.store(x.load()),
        x.store(y.load()),
        y.store(z.load()),
    )
```

Calling a subroutine: swap(l,m)

Source: https://pyteal.readthedocs.io/en/stable/control_structures.html

Recursion



```
@Subroutine(TealType.uint64)
def recursiveIsEven(i):
    return (
        If(i == Int(0))
        .Then(Int(1))
        .ElseIf(i == Int(1))
        .Then(Int(0))
        .Else(recursiveIsEven(i - Int(2)))
    )
```

Calling a function recursively: recursiveIsEven(l,m)

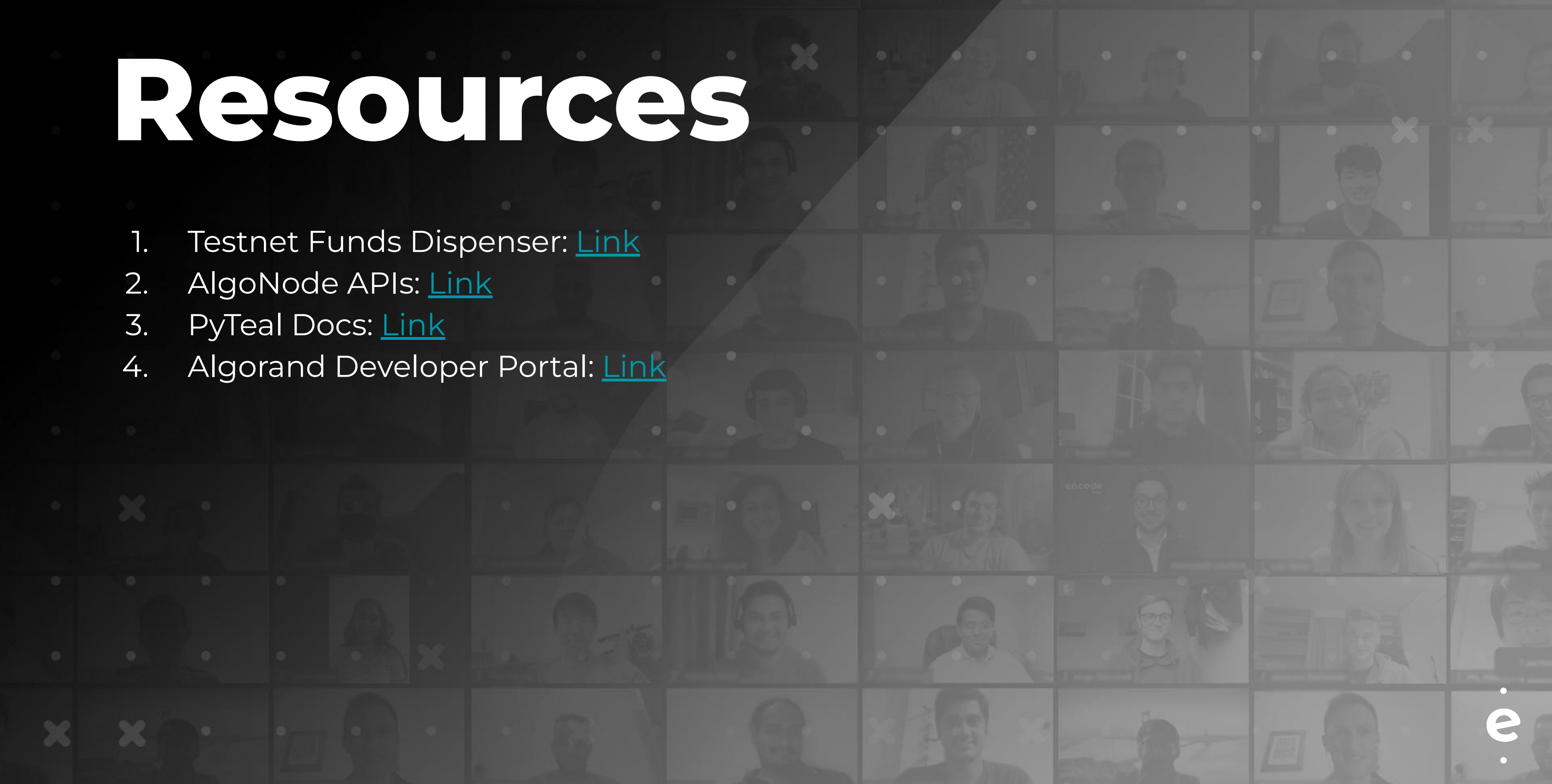
Source: https://pyteal.readthedocs.io/en/stable/control_structures.html

Assignment

1. Review the slides and recordings of this week
2. Two parter: Coding assignment, and a google forms quiz (will be shared tomorrow)
3. Coding assignment:
    a. Create an ASA - with ENB as the symbol
    b. Deploy a smart contract and store the ASA ID created in step 3.a in the global state when you deploy the application
    c. Build upon the voting application with the following requirements:
        i. Voters must hold a minimum of 1000 ENB when voting
        ii. Vote Method must only accept the vote choice of “yes”, “no”, “abstain”. Passing any other choice must automatically be rejected by logic
        iii. Vote count must be increased by the amount of ENB the voter holds at the time of voting (must be decreased by the same vote amount if voter opts out/clear state/close out)

Resources

1. Testnet Funds Dispenser: Link
2. AlgoNode APIs: Link
3. PyTeal Docs: Link
4. Algorand Developer Portal: Link



Thank You

Email: hi@lalithmedury.com

Twitter: @LalithMedury

Web: https://lalithmedury.com

Email: info@encode.club

Twitter: @encodeclub

Web: https://encode.club