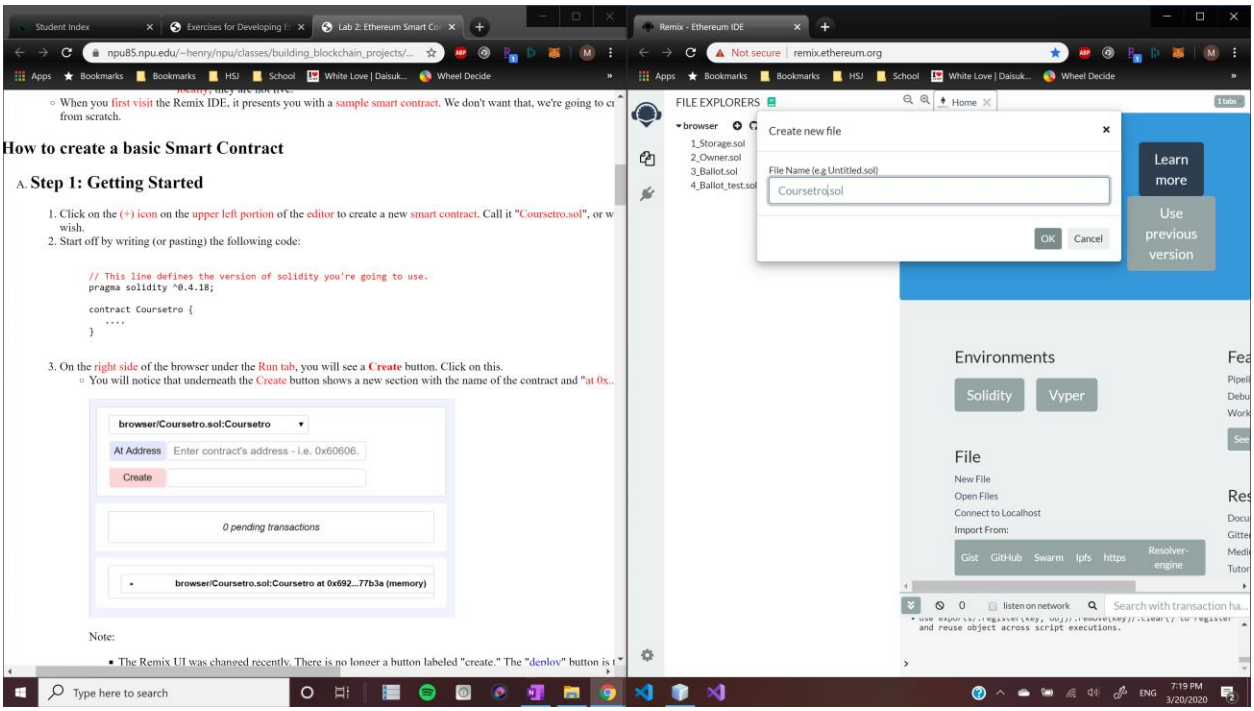
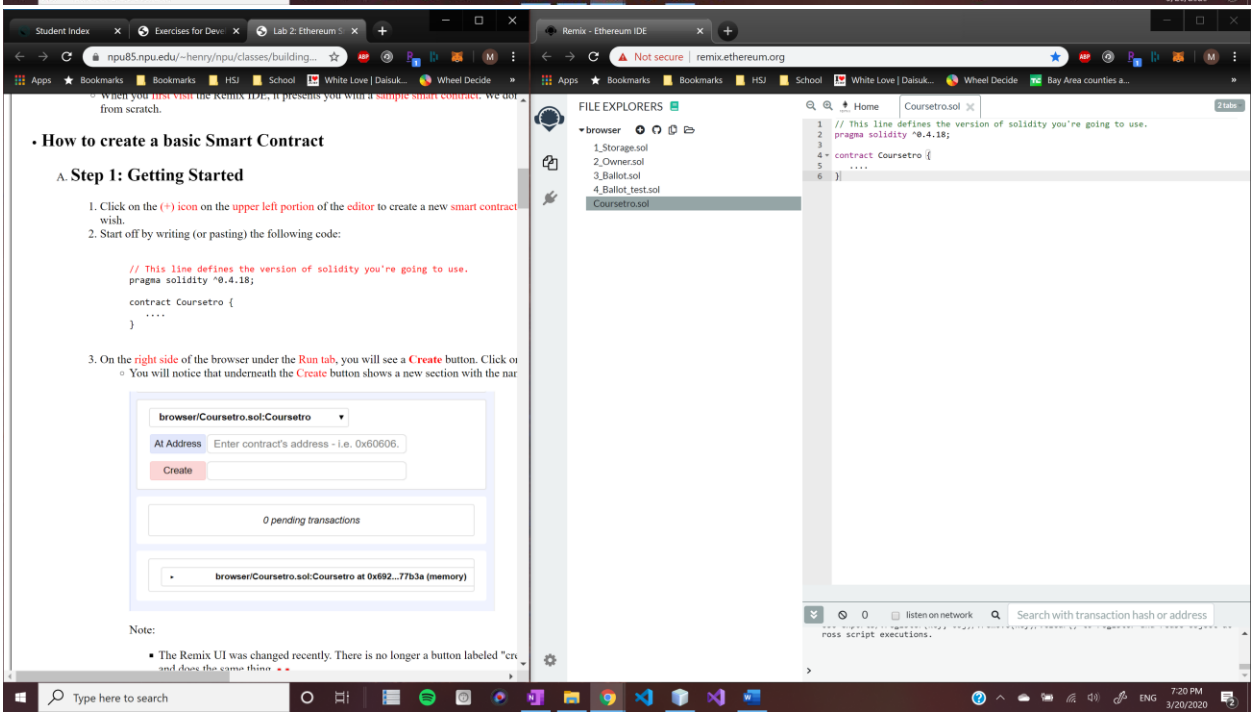
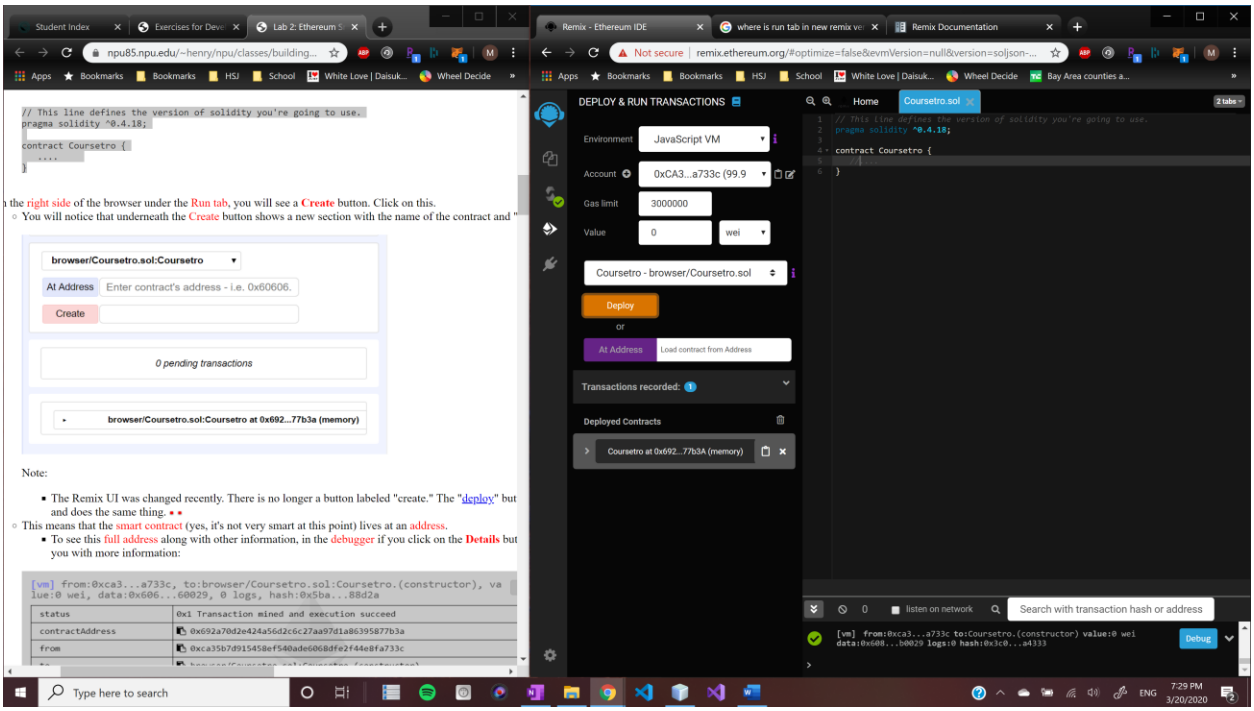
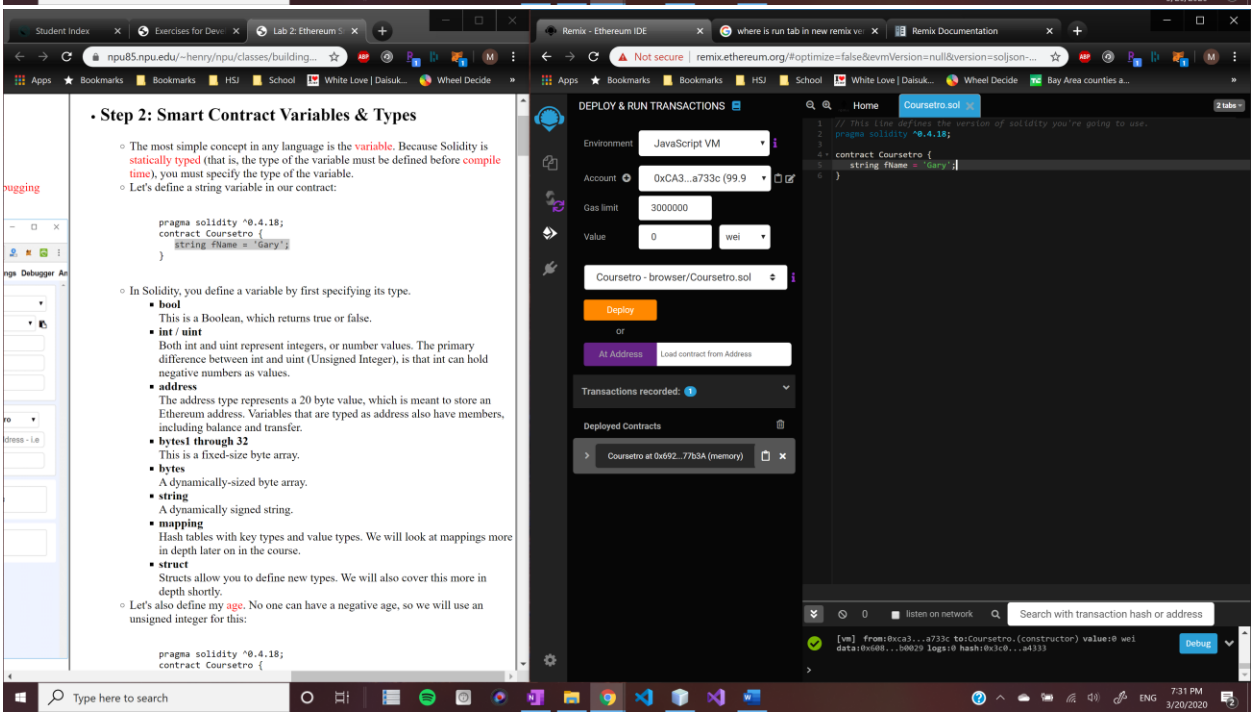
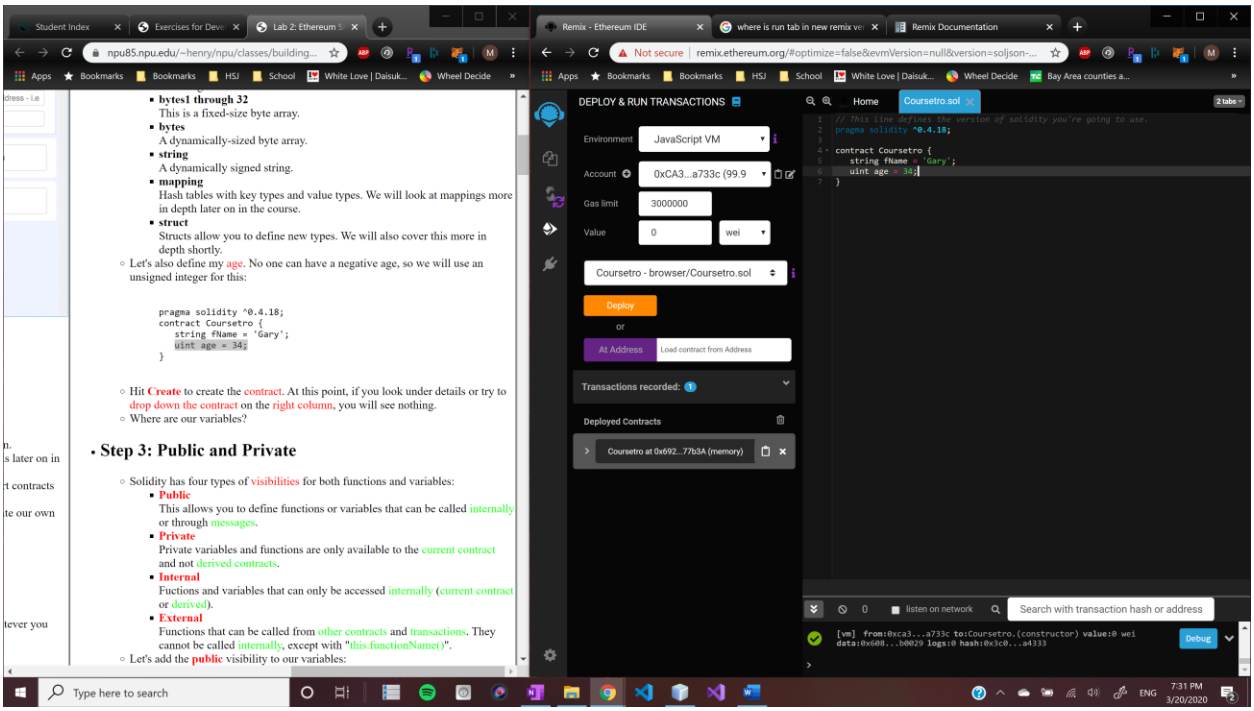
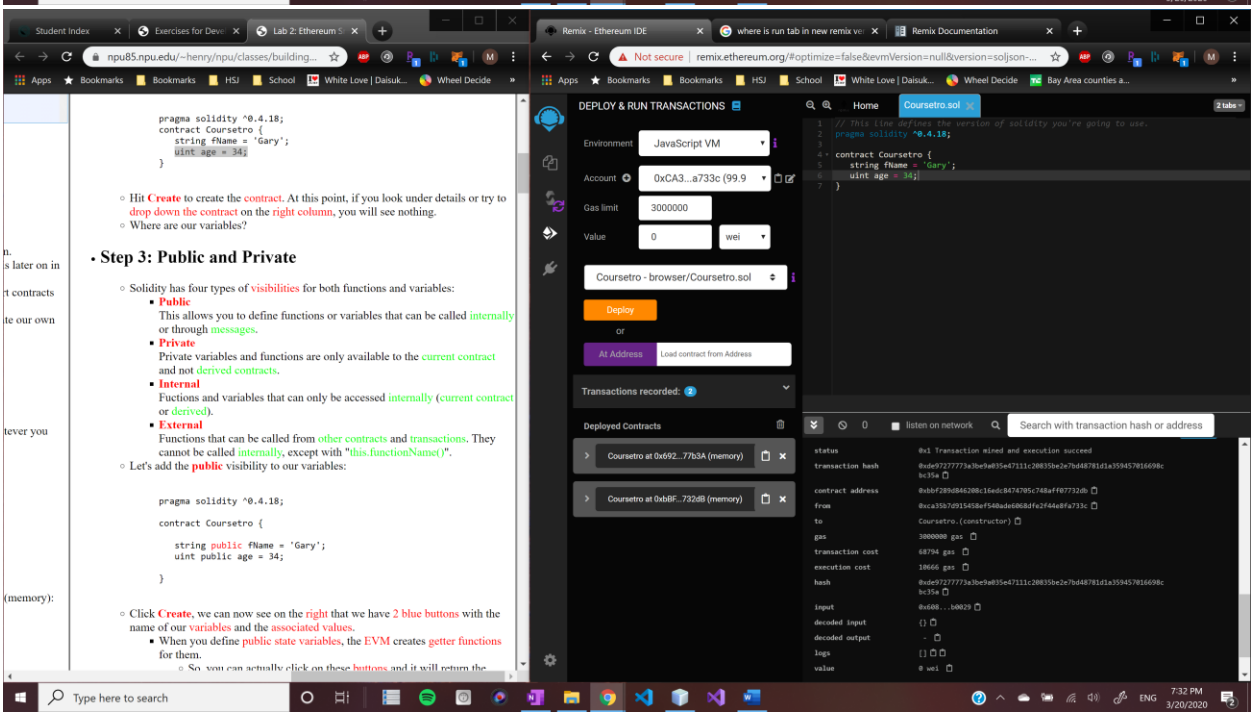

Student Index
Exercises for Developing
Lab 2: Ethereum Smart Co
npu85.npu.edu/~henry/npu/classes/building_blockchain_projects/...
Apps
Bookmarks
HSJ
School
White Love | Daisuk...
Wheel Decide
When you first visit the Remix IDE, it presents you with a sample smart contract. We don't want that, we're going to cr
from scratch.
How to create a basic Smart Contract
A. Step 1: Getting Started
1. Click on the (+) icon on the upper left portion of the editor to create a new smart contract. Call it "Coursetro.sol", or w
wish.
2. Start off by writing (or pasting) the following code:
// This line defines the version of solidity you're going to use.
pragma solidity ^0.4.18;
contract Coursetro {
....
}
3. On the right side of the browser under the Run tab, you will see a Create button. Click on this.
You will notice that underneath the Create button shows a new section with the name of the contract and "at 0x..
browser/Coursetro.sol:Coursetro
At Address
Enter contract's address - i.e. 0x60606.
Create
0 pending transactions
browser/Coursetro.sol:Coursetro at 0x692...77b3a (memory)
Note:
The Remix UI was changed recently. There is no longer a button labeled "create." The "deploy" button is
Remix - Ethereum IDE
Not secure
remix.ethereum.org
FILE EXPLORERS
browser
1_Storage.sol
2_Owner.sol
3_Ballot.sol
4_Ballot_test.sol
Create new file
File Name (e.g Untitled.sol)
Coursetro.sol
OK
Cancel
Learn more
Use previous version
Environments
Solidity
Vyper
File
New File
Open Files
Connect to Localhost
Import From:
Gist
GitHub
Swarm
Ipfs
https
Resolver-engine
listen on network
Search with transaction ha...
and reuse object across script executions.
Type here to search
ENG
7:19 PM
3/20/2020

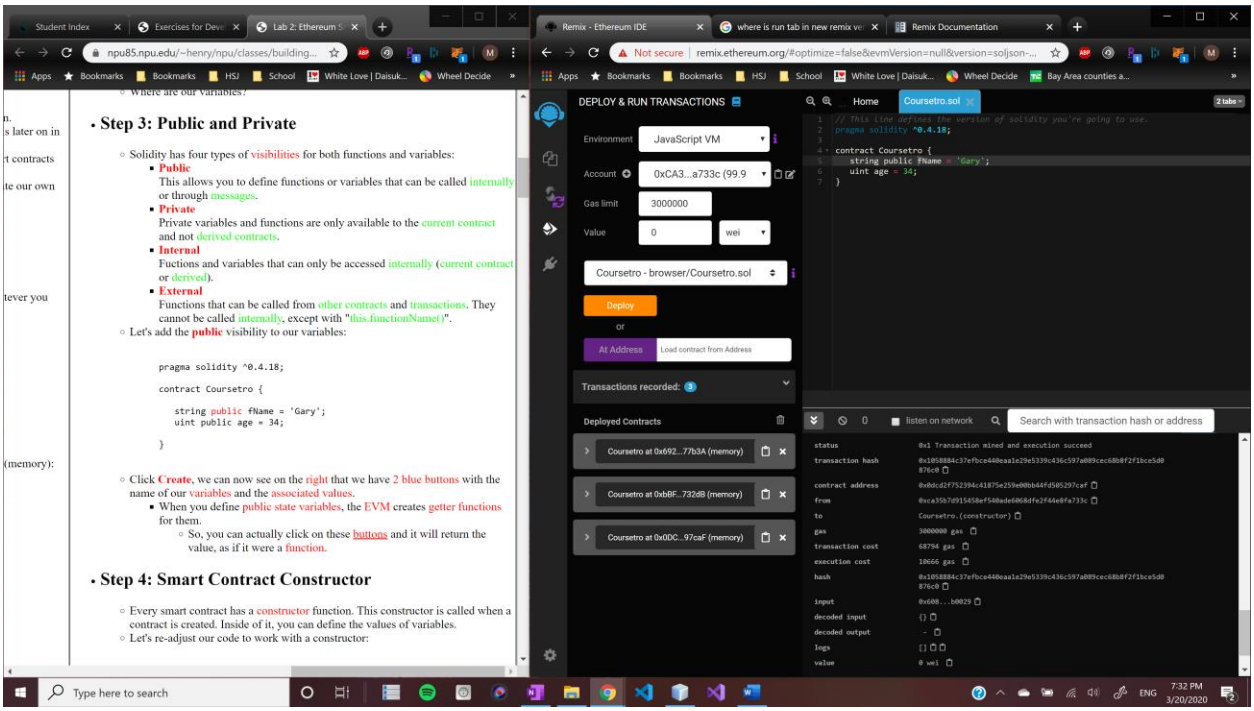

Student Index
Exercises for Deve
Lab 2: Ethereum S
npu85.npu.edu/~henry/npu/classes/building...
Apps
Bookmarks
HSJ
School
White Love | Daisuk...
Wheel Decide
When you first visit the Remix IDE, it presents you with a sample smart contract. We don
from scratch.
How to create a basic Smart Contract
A. Step 1: Getting Started
1. Click on the (+) icon on the upper left portion of the editor to create a new smart contract
wish.
2. Start off by writing (or pasting) the following code:
// This line defines the version of solidity you're going to use.
pragma solidity ^0.4.18;
contract Coursetro {
....
}
3. On the right side of the browser under the Run tab, you will see a Create button. Click o
You will notice that underneath the Create button shows a new section with the nar
browser/Coursetro.sol:Coursetro
At Address
Enter contract's address - i.e. 0x60606.
Create
0 pending transactions
browser/Coursetro.sol:Coursetro at 0x692...77b3a (memory)
Note:
The Remix UI was changed recently. There is no longer a button labeled "cr
and does the same thing
Remix - Ethereum IDE
Not secure
remix.ethereum.org
Bay Area counties a...
FILE EXPLORERS
browser
1_Storage.sol
2_Owner.sol
3_Ballot.sol
4_Ballot_test.sol
Coursetro.sol
Home
Coursetro.sol
2 tabs
// This line defines the version of solidity you're going to use.
pragma solidity ^0.4.18;
contract Coursetro {
....
}
listen on network
Search with transaction hash or address
ross script executions.
Type here to search
ENG
7:20 PM
3/20/2020

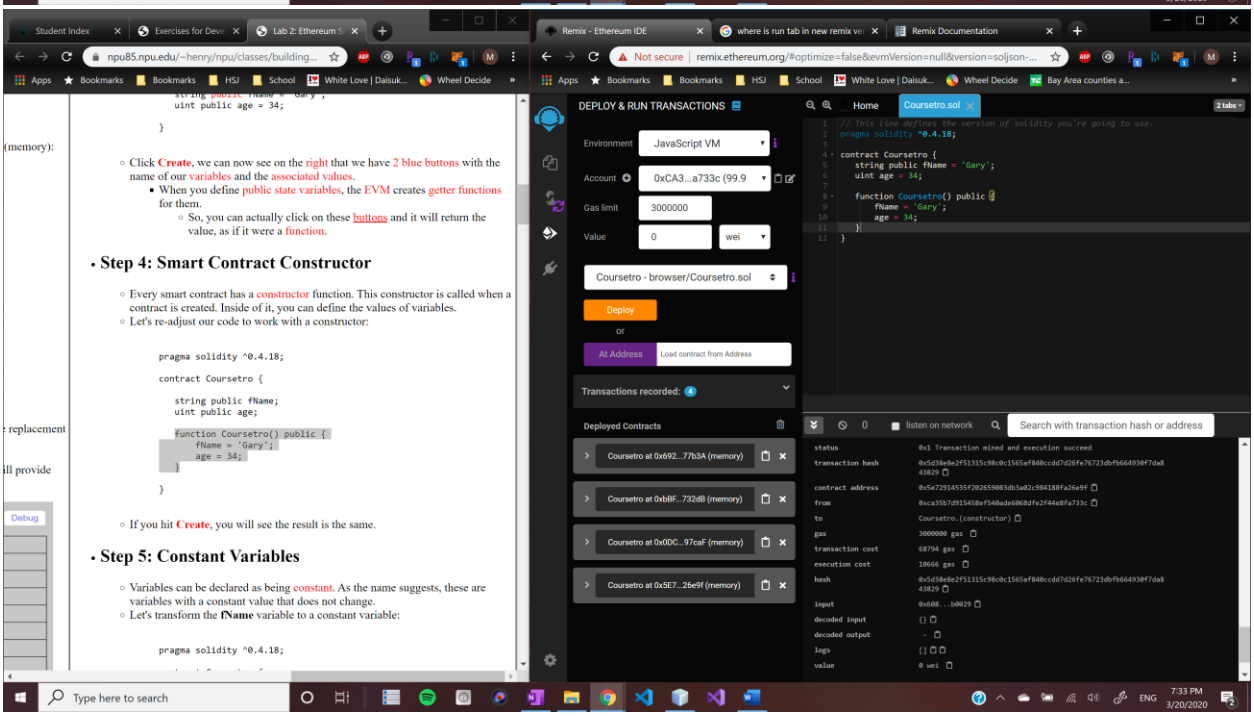

Student Index
Exercises for Deve
Lab 2: Ethereum S
npu85.npu.edu/~henry/npu/classes/building...
Apps
Bookmarks
Bookmarks
HSJ
School
White Love | Daisuk...
Wheel Decide
// This line defines the version of solidity you're going to use.
pragma solidity ^0.4.18;
contract Coursetro {
....
}
the right side of the browser under the Run tab, you will see a Create button. Click on this.
You will notice that underneath the Create button shows a new section with the name of the contract and "
browser/Coursetro.sol:Coursetro
At Address
Enter contract's address - i.e. 0x60606.
Create
0 pending transactions
browser/Coursetro.sol:Coursetro at 0x692...77b3a (memory)
Note:
The Remix UI was changed recently. There is no longer a button labeled "create." The "deploy" but and does the same thing.
This means that the smart contract (yes, it's not very smart at this point) lives at an address.
To see this full address along with other information, in the debugger if you click on the Details but you with more information:
[vm] from:0xca3...a733c, to:browser/Coursetro.sol:Coursetro.(constructor), value:0 wei, data:0x606...60029, 0 logs, hash:0x5ba...88d2a
status
0x1 Transaction mined and execution succeed
contractAddress
0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
from
0xca35b7d915458ef540ade6068dfe2f44e8fa733c
Remix - Ethereum IDE
where is run tab in new remix ve
Remix Documentation
Not secure
remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-...
DEPLOY & RUN TRANSACTIONS
Environment
JavaScript VM
Account
0xCA3...a733c (99.9
Gas limit
3000000
Value
0
wei
Coursetro - browser/Coursetro.sol
Deploy
or
At Address
Load contract from Address
Transactions recorded:
Deployed Contracts
Coursetro at 0x692...77b3A (memory)
Home
Coursetro.sol
2 tabs
// This line defines the version of solidity you're going to use.
pragma solidity ^0.4.18;
contract Coursetro {
}
listen on network
Search with transaction hash or address
[vm] from:0xca3...a733c to:Coursetro.(constructor) value:0 wei data:0x608...b0029 logs:0 hash:0x3c0...a4333
Debug
Type here to search
ENG
7:29 PM
3/20/2020

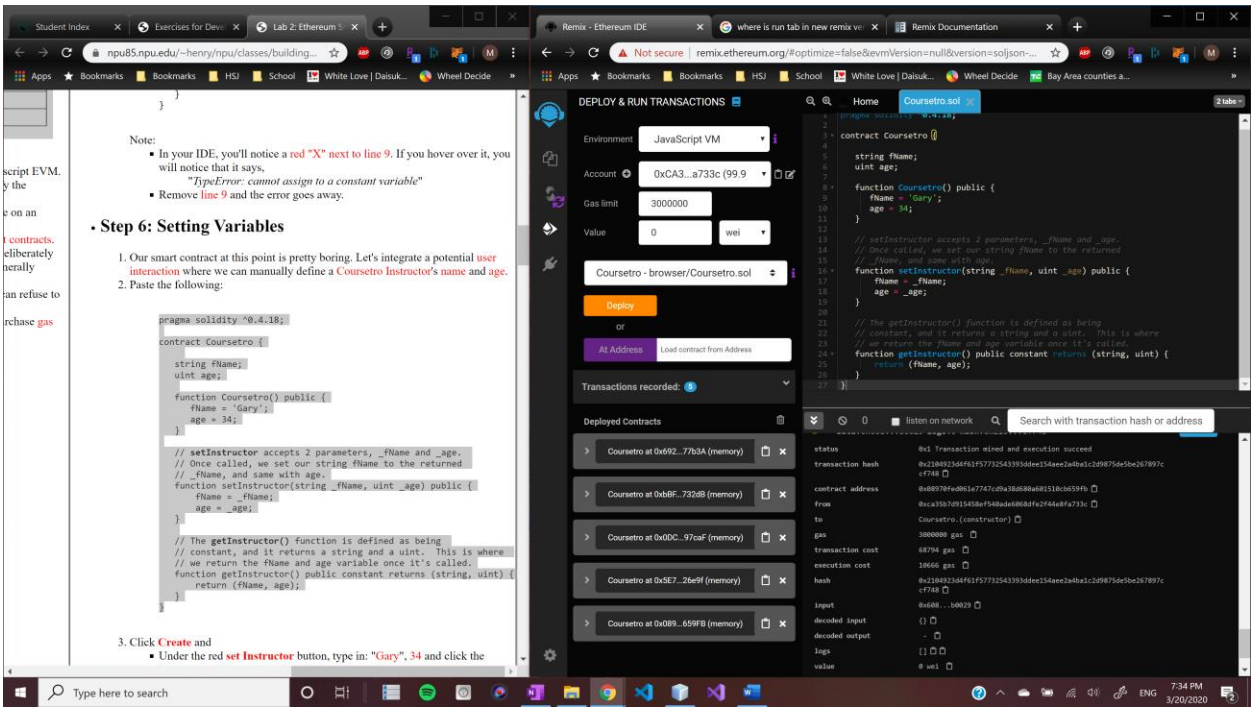

Student Index
Exercises for Deve
Lab 2: Ethereum S
npu85.npu.edu/~henry/npu/classes/building...
Apps
Bookmarks
Bookmarks
HSJ
School
White Love | Daisuk...
Wheel Decide
Step 2: Smart Contract Variables & Types
The most simple concept in any language is the variable. Because Solidity is statically typed (that is, the type of the variable must be defined before compile time), you must specify the type of the variable.
Let's define a string variable in our contract:
pragma solidity ^0.4.18;
contract Coursetro {
string fName = 'Gary';
}
In Solidity, you define a variable by first specifying its type.
bool
This is a Boolean, which returns true or false.
int / uint
Both int and uint represent integers, or number values. The primary difference between int and uint (Unsigned Integer), is that int can hold negative numbers as values.
address
The address type represents a 20 byte value, which is meant to store an Ethereum address. Variables that are typed as address also have members, including balance and transfer.
bytes1 through 32
This is a fixed-size byte array.
bytes
A dynamically-sized byte array.
string
A dynamically signed string.
mapping
Hash tables with key types and value types. We will look at mappings more in depth later on in the course.
struct
Structs allow you to define new types. We will also cover this more in depth shortly.
Let's also define my age. No one can have a negative age, so we will use an unsigned integer for this:
pragma solidity ^0.4.18;
contract Coursetro {
Remix - Ethereum IDE
where is run tab in new remix ve
Remix Documentation
Not secure
remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-...
DEPLOY & RUN TRANSACTIONS
Environment
JavaScript VM
Account
0xCA3...a733c (99.9
Gas limit
3000000
Value
0
wei
Coursetro - browser/Coursetro.sol
Deploy
or
At Address
Load contract from Address
Transactions recorded:
Deployed Contracts
Coursetro at 0x692...77b3A (memory)
Home
Coursetro.sol
2 tabs
// This line defines the version of solidity you're going to use.
pragma solidity ^0.4.18;
contract Coursetro {
string fName = 'Gary';
}
listen on network
Search with transaction hash or address
[vm] from:0xca3...a733c to:Coursetro.(constructor) value:0 wei data:0x608...b0029 logs:0 hash:0x3c0...a4333
Debug
Type here to search
ENG
7:31 PM
3/20/2020


Student Index
Exercises for Deve
Lab 2: Ethereum S
npu85.npu.edu/~henry/npu/classes/building...
Apps
Bookmarks
Bookmarks
HSJ
School
White Love | Daisuk...
Wheel Decide
bytes1 through 32
This is a fixed-size byte array.
bytes
A dynamically-sized byte array.
string
A dynamically signed string.
mapping
Hash tables with key types and value types. We will look at mappings more in depth later on in the course.
struct
Structs allow you to define new types. We will also cover this more in depth shortly.
Let's also define my age. No one can have a negative age, so we will use an unsigned integer for this:
pragma solidity ^0.4.18;
contract Coursetro {
string fName = 'Gary';
uint age = 34;
}
Hit Create to create the contract. At this point, if you look under details or try to drop down the contract on the right column, you will see nothing.
Where are our variables?
Step 3: Public and Private
Solidity has four types of visibilities for both functions and variables:
Public
This allows you to define functions or variables that can be called internally or through messages.
Private
Private variables and functions are only available to the current contract and not derived contracts.
Internal
Fuctions and variables that can only be accessed internally (current contract or derived).
External
Functions that can be called from other contracts and transactions. They cannot be called internally, except with "this.functionName()".
Let's add the public visibility to our variables:
Type here to search
Remix - Ethereum IDE
where is run tab in new remix ve
Remix Documentation
Not secure
remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-...
DEPLOY & RUN TRANSACTIONS
Environment
JavaScript VM
Account
0xCA3...a733c (99.9
Gas limit
3000000
Value
0
wei
Coursetro - browser/Coursetro.sol
Deploy
or
At Address
Load contract from Address
Transactions recorded: 1
Deployed Contracts
Coursetro at 0x692...77b3A (memory)
Home
Coursetro.sol
2 tabs
// This line defines the version of solidity you're going to use.
pragma solidity ^0.4.18;
contract Coursetro {
string fName = 'Gary';
uint age = 34;
}
listen on network
Search with transaction hash or address
[vm] from:0xca3...a733c to:Coursetro.(constructor) value:0 wei data:0x608...b0029 logs:0 hash:0x3c0...a4333
Debug
ENG
7:31 PM
3/20/2020


Student Index
Exercises for Deve
Lab 2: Ethereum S
npu85.npu.edu/~henry/npu/classes/building...
Apps
Bookmarks
Bookmarks
HSJ
School
White Love | Daisuk...
Wheel Decide
pragma solidity ^0.4.18;
contract Coursetro {
string fName = 'Gary';
uint age = 34;
}
Hit Create to create the contract. At this point, if you look under details or try to drop down the contract on the right column, you will see nothing.
Where are our variables?
Step 3: Public and Private
Solidity has four types of visibilities for both functions and variables:
Public
This allows you to define functions or variables that can be called internally or through messages.
Private
Private variables and functions are only available to the current contract and not derived contracts.
Internal
Fuctions and variables that can only be accessed internally (current contract or derived).
External
Functions that can be called from other contracts and transactions. They cannot be called internally, except with "this.functionName()".
Let's add the public visibility to our variables:
pragma solidity ^0.4.18;
contract Coursetro {
string public fName = 'Gary';
uint public age = 34;
}
Click Create, we can now see on the right that we have 2 blue buttons with the name of our variables and the associated values.
When you define public state variables, the EVM creates getter functions for them.
So, you can actually click on these buttons and it will return the
Type here to search
Remix - Ethereum IDE
where is run tab in new remix ve
Remix Documentation
Not secure
remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-...
DEPLOY & RUN TRANSACTIONS
Environment
JavaScript VM
Account
0xCA3...a733c (99.9
Gas limit
3000000
Value
0
wei
Coursetro - browser/Coursetro.sol
Deploy
or
At Address
Load contract from Address
Transactions recorded: 2
Deployed Contracts
Coursetro at 0x692...77b3A (memory)
Coursetro at 0xbBF...732d8 (memory)
Home
Coursetro.sol
2 tabs
// This line defines the version of solidity you're going to use.
pragma solidity ^0.4.18;
contract Coursetro {
string fName = 'Gary';
uint age = 34;
}
listen on network
Search with transaction hash or address
status
0x1 Transaction mined and execution succeed
transaction hash
0xde97277773a3be9a835e47111c20835be2e7bd48781d1a35945701698c
bc35a
contract address
0xbbf289d846208c16edc8474705c748aff07732db
from
0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to
Coursetro.(constructor)
gas
3000000 gas
transaction cost
68794 gas
execution cost
10666 gas
hash
0xde97277773a3be9a835e47111c20835be2e7bd48781d1a35945701698c
bc35a
input
0x608...b0029
decoded input
{}
decoded output
-
logs
[]
value
0 wei
ENG
7:32 PM
3/20/2020


Student Index
Exercises for Deve
Lab 2: Ethereum S
npu85.npu.edu/~henry/npu/classes/building...
Apps
Bookmarks
HSJ
School
White Love | Daisuk...
Wheel Decide
Where are our variables?
Step 3: Public and Private
Solidity has four types of visibilities for both functions and variables:
Public
This allows you to define functions or variables that can be called internally or through messages.
Private
Private variables and functions are only available to the current contract and not derived contracts.
Internal
Fuctions and variables that can only be accessed internally (current contract or derived).
External
Functions that can be called from other contracts and transactions. They cannot be called internally, except with "this.functionName()".
Let's add the public visibility to our variables:
pragma solidity ^0.4.18;
contract Coursetro {
string public fName = 'Gary';
uint public age = 34;
}
Click Create, we can now see on the right that we have 2 blue buttons with the name of our variables and the associated values.
When you define public state variables, the EVM creates getter functions for them.
So, you can actually click on these buttons and it will return the value, as if it were a function.
Step 4: Smart Contract Constructor
Every smart contract has a constructor function. This constructor is called when a contract is created. Inside of it, you can define the values of variables.
Let's re-adjust our code to work with a constructor:
Remix - Ethereum IDE
where is run tab in new remix ve
Remix Documentation
Not secure
remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-...
DEPLOY & RUN TRANSACTIONS
Environment
JavaScript VM
Account
0xCA3...a733c (99.9
Gas limit
3000000
Value
0
wei
Coursetro - browser/Coursetro.sol
Deploy
or
At Address
Load contract from Address
Transactions recorded: 3
Deployed Contracts
Coursetro at 0x692...77b3A (memory)
Coursetro at 0xbBF...732d8 (memory)
Coursetro at 0x0DC...97caF (memory)
Home
Coursetro.sol
2 tabs
// This line defines the version of solidity you're going to use.
pragma solidity ^0.4.18;
contract Coursetro {
string public fName = 'Gary';
uint age = 34;
}
listen on network
Search with transaction hash or address
status
0x1 Transaction mined and execution succeed
transaction hash
0x1058884c37efbce440eaa1e29e5339c436c597a089cec68b8f2f1bce5d0876c0
contract address
0x0dcd2f752394c41875e259e00bb44fd505297caf
from
0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to
Coursetro.(constructor)
gas
3000000 gas
transaction cost
68794 gas
execution cost
10666 gas
hash
0x1058884c37efbce440eaa1e29e5339c436c597a089cec68b8f2f1bce5d0876c0
input
0x608...b0029
decoded input
{}
decoded output
-
logs
[]
value
0 wei
Type here to search
ENG
7:32 PM
3/20/2020


Student Index
Exercises for Deve
Lab 2: Ethereum S
npu85.npu.edu/~henry/npu/classes/building...
Apps
Bookmarks
HSJ
School
White Love | Daisuk...
Wheel Decide
uint public age = 34;
}
Click Create, we can now see on the right that we have 2 blue buttons with the name of our variables and the associated values.
When you define public state variables, the EVM creates getter functions for them.
So, you can actually click on these buttons and it will return the value, as if it were a function.
Step 4: Smart Contract Constructor
Every smart contract has a constructor function. This constructor is called when a contract is created. Inside of it, you can define the values of variables.
Let's re-adjust our code to work with a constructor:
pragma solidity ^0.4.18;
contract Coursetro {
string public fName;
uint public age;
function Coursetro() public {
fName = 'Gary';
age = 34;
}
}
If you hit Create, you will see the result is the same.
Step 5: Constant Variables
Variables can be declared as being constant. As the name suggests, these are variables with a constant value that does not change.
Let's transform the fName variable to a constant variable:
pragma solidity ^0.4.18;
Debug
Remix - Ethereum IDE
where is run tab in new remix ve
Remix Documentation
Not secure
remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-...
DEPLOY & RUN TRANSACTIONS
Environment
JavaScript VM
Account
0xCA3...a733c (99.9
Gas limit
3000000
Value
0
wei
Coursetro - browser/Coursetro.sol
Deploy
or
At Address
Load contract from Address
Transactions recorded: 4
Deployed Contracts
Coursetro at 0x692...77b3A (memory)
Coursetro at 0xbBF...732d8 (memory)
Coursetro at 0x0DC...97caF (memory)
Coursetro at 0x5E7...26e9f (memory)
Home
Coursetro.sol
2 tabs
// This line defines the version of solidity you're going to use.
pragma solidity ^0.4.18;
contract Coursetro {
string public fName = 'Gary';
uint age = 34;
function Coursetro() public {
fName = 'Gary';
age = 34;
}
}
listen on network
Search with transaction hash or address
status
0x1 Transaction mined and execution succeed
transaction hash
0x5d38e8e2f51315c98c0c1565af840ccdd7d26fe76723dbfb664930ff7da843829
contract address
0x5e72914535f202659083db3a02c984188fa26e9f
from
0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to
Coursetro.(constructor)
gas
3000000 gas
transaction cost
68794 gas
execution cost
10666 gas
hash
0x5d38e8e2f51315c98c0c1565af840ccdd7d26fe76723dbfb664930ff7da843829
input
0x608...b0029
decoded input
{}
decoded output
-
logs
[]
value
0 wei
Type here to search
ENG
7:33 PM
3/20/2020


Student Index
Exercises for Deve
Lab 2: Ethereum S
npu85.npu.edu/~henry/npu/classes/building...
Apps
Bookmarks
Bookmarks
HSJ
School
White Love | Daisuk...
Wheel Decide
}
}
Note:
In your IDE, you'll notice a red "X" next to line 9. If you hover over it, you will notice that it says,
"TypeError: cannot assign to a constant variable"
Remove line 9 and the error goes away.
Step 6: Setting Variables
1. Our smart contract at this point is pretty boring. Let's integrate a potential user interaction where we can manually define a Coursetro Instructor's name and age.
2. Paste the following:
pragma solidity ^0.4.18;
contract Coursetro {
string fName;
uint age;
function Coursetro() public {
fName = 'Gary';
age = 34;
}
// setInstructor accepts 2 parameters, _fName and _age.
// Once called, we set our string fName to the returned
// _fName, and same with age.
function setInstructor(string _fName, uint _age) public {
fName = _fName;
age = _age;
}
// The getInstructor() function is defined as being
// constant, and it returns a string and a uint. This is where
// we return the fName and age variable once it's called.
function getInstructor() public constant returns (string, uint) {
return (fName, age);
}
}
3. Click Create and
Under the red set Instructor button, type in: "Gary", 34 and click the
Type here to search
Remix - Ethereum IDE
where is run tab in new remix ve
Remix Documentation
Not secure
remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-...
DEPLOY & RUN TRANSACTIONS
Environment
JavaScript VM
Account
0xCA3...a733c (99.9
Gas limit
3000000
Value
0
wei
Coursetro - browser/Coursetro.sol
Deploy
or
At Address
Load contract from Address
Transactions recorded: 5
Deployed Contracts
Coursetro at 0x692...77b3A (memory)
Coursetro at 0xbBF...732d8 (memory)
Coursetro at 0x0DC...97caF (memory)
Coursetro at 0x5E7...26e9f (memory)
Coursetro at 0x0B9...659FB (memory)
Home
Coursetro.sol
2 tabs
contract Coursetro {
string fName;
uint age;
function Coursetro() public {
fName = 'Gary';
age = 34;
}
// setInstructor accepts 2 parameters, _fName and _age.
// Once called, we set our string fName to the returned
// _fName, and same with age.
function setInstructor(string _fName, uint _age) public {
fName = _fName;
age = _age;
}
// The getInstructor() function is defined as being
// constant, and it returns a string and a uint. This is where
// we return the fName and age variable once it's called.
function getInstructor() public constant returns (string, uint) {
return (fName, age);
}
}
listen on network
Search with transaction hash or address
status
0x1 Transaction mined and execution succeed
transaction hash
0x2104923d4f61f5773254339 3ddee154aee2a4ba1c2d9875de5be267897c cf748
contract address
0x08970fed861e7747cd9a38d680a681510cb659fb
from
0xca35b7d915458ef540ade6068dfe2f44e8fa733c
to
Coursetro.(constructor)
gas
3000000 gas
transaction cost
68794 gas
execution cost
10666 gas
hash
0x2104923d4f61f5773254339 3ddee154aee2a4ba1c2d9875de5be267897c cf748
input
0x608...b0029
decoded input
{}
decoded output
-
logs
[]
value
0 wei
ENG
7:34 PM
3/20/2020