

“Code Your Own Cryptocurrency on Ethereum (ERC-20 Token)”

Note: This document is made for educational reference only.

Credit: Learning from Udemy course "Code Your Own Cryptocurrency on Ethereum (ERC-20 Token)"

Reference Website: <https://www.udemy.com/course/code-your-own-cryptocurrency/>

Learn Outcome:

- To build an ERC-20 token and sell it with a crowd sale (ICO) website.
- Code your first smart contracts in Solidity.
- Deploy the smart contracts to the blockchain.
- Learn essential Ethereum development tools and best practices, like running your own Etherum node with Geth and testing Smart Contracts.

Outline:

Session 1: Basic Installation

Session 2: Project Setup & Smoke Test

Session 3: Creating the ERC-20 Token Smart Contract

Session 4: Transferring Tokens

Session 5: Approving Token Transfers

Session 6: Delegated Token Transfers

Session 7: Console Demonstration

Session 8: Creating the Crowd Sale Smart Contract

Session 9: Purchasing ERC-20 Tokens

Session 10: Ending Token Sale

Session 11: Creating the Crowd Sale Website

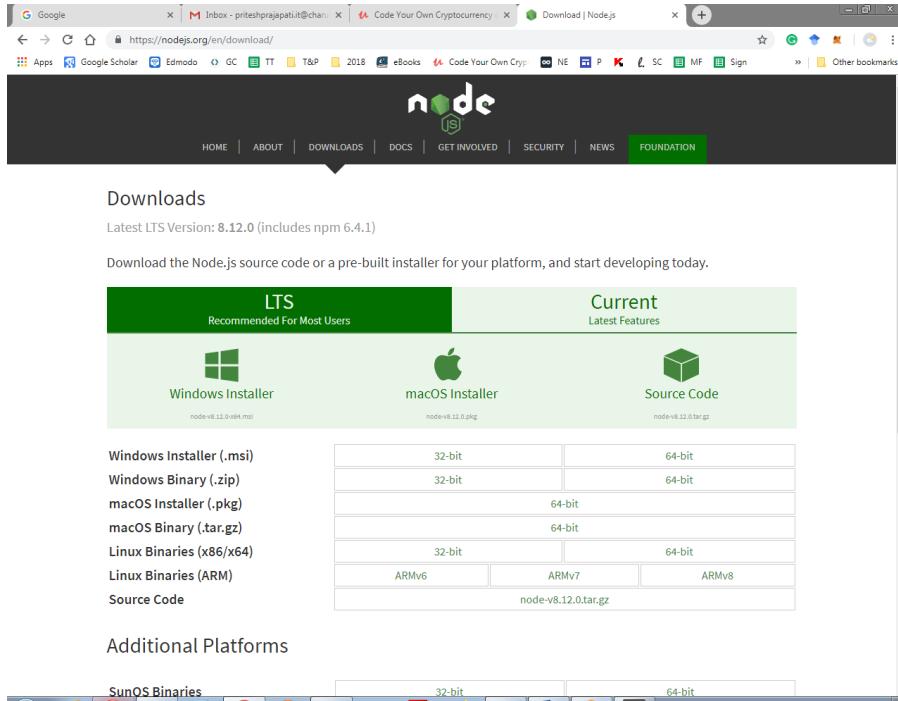
Session 12: Building out the Crowd Sale Website

Session 13: Purchasing ERC-20 Tokens

Session 14: Deploying the Smart Contracts with Geth (**Not Fully Completed**)

Session 1: Basic Installation

- Download the Node.js source code or a pre-built installer for your platform
<https://nodejs.org/en/download/>

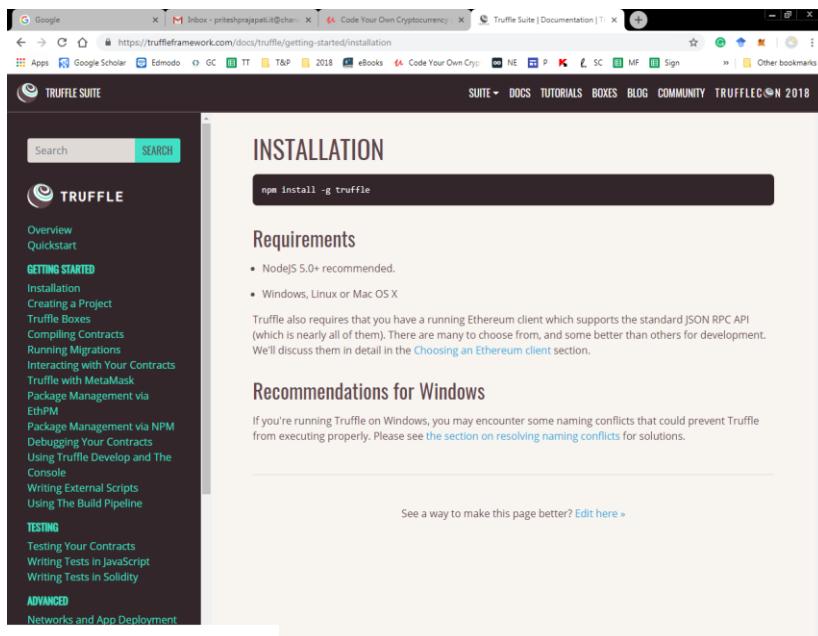


The screenshot shows the Node.js download page. At the top, there are two tabs: 'LTS Recommended For Most Users' and 'Current Latest Features'. Under 'LTS', there are links for 'Windows Installer (.msi)', 'Windows Binary (.zip)', 'macOS Installer (.pkg)', 'macOS Binary (.tar.gz)', 'Linux Binaries (x86/x64)', 'Linux Binaries (ARM)', and 'Source Code'. Under 'Current', there are links for 'Windows 32-bit', 'Windows 64-bit', 'macOS 32-bit', 'macOS 64-bit', 'Linux 32-bit', 'Linux 64-bit', 'ARMv6', 'ARMv7', and 'ARMv8'. Below these sections, there is a 'Additional Platforms' section with a 'SunOS Binaries' link.

- Install TRUFFLE

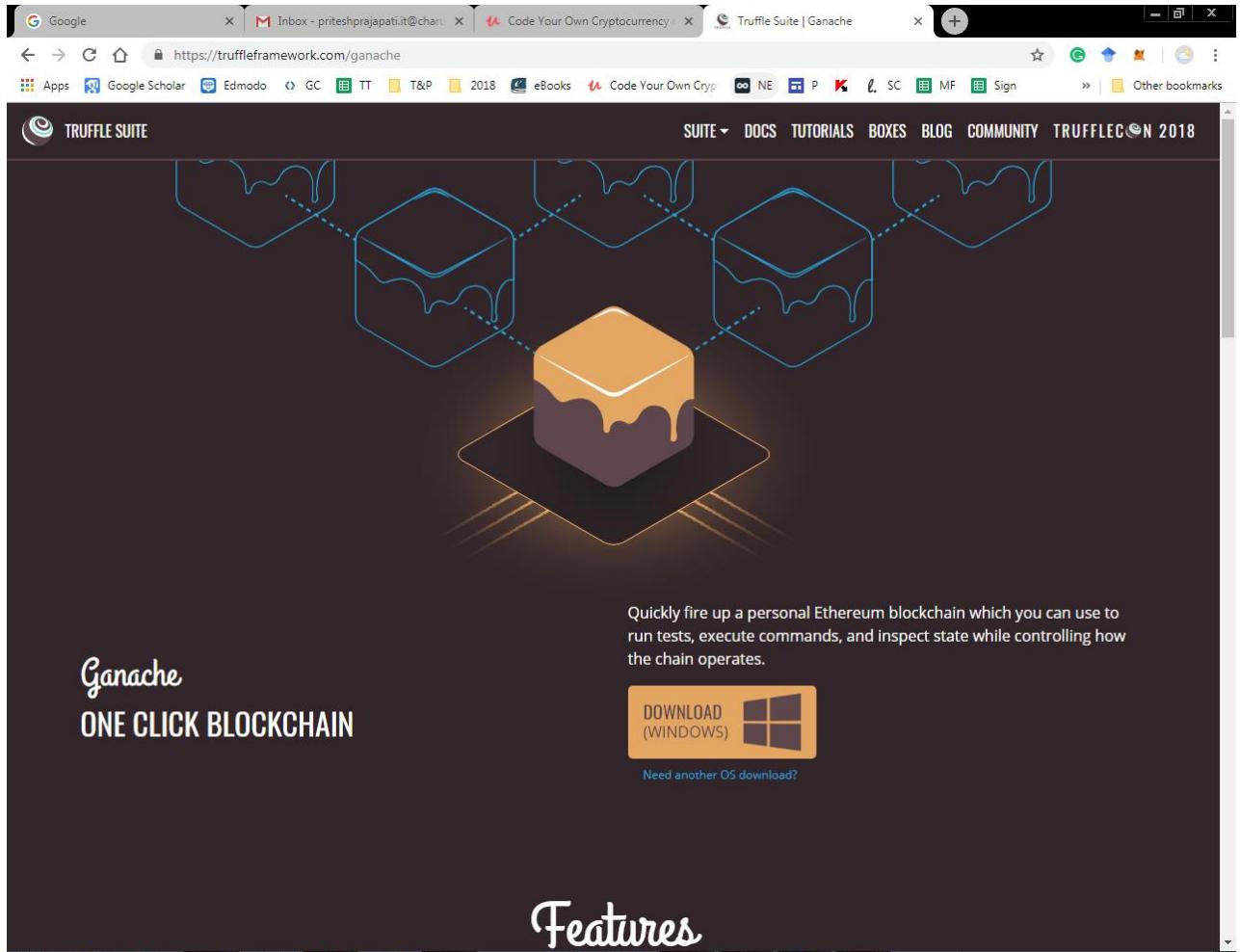
`npm install -g truffle`

<https://truffleframework.com/docs/truffle/getting-started/installation>

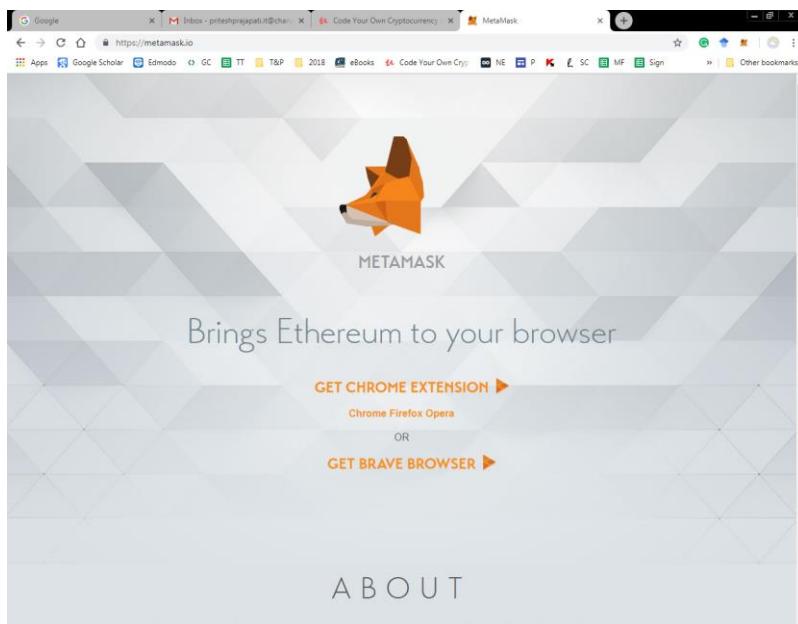


The screenshot shows the Truffle Suite documentation page under the 'INSTALLATION' section. It features a search bar at the top. Below it, there is a command line interface (CLI) input field containing the command `npm install -g truffle`. The page is divided into several sections: 'Requirements' (Node.js 5.0+, Windows, Linux or Mac OS X), 'Recommendations for Windows' (warning about naming conflicts), and a footer with a link to edit the page.

3. Install GANACHE <https://truffleframework.com/ganache>



4. Install METAMASK for your browser <https://metamask.io/>



5. Install Package Control Ethereum <https://packagecontrol.io/packages/Ethereum>

The screenshot shows a web browser window with the URL <https://packagecontrol.io/packages/Ethereum>. The page title is "Package Control". On the left, there's a "BROWSE" section for "Ethereum" by davidhq, labeled ST2/ST3. It includes a "language syntax" label. Below this is a "Details" section with version 1.0.3, homepage github.com, issues github.com, modified 1 month ago, last seen 1 hour ago, and first seen 4 years ago. To the right is a "Installs" chart titled "TOTAL 24K" showing daily installs from Oct 24 to Sep 12, with segments for WIN (blue), OS X (light blue), and LINUX (purple). A sidebar on the right lists links like Installation, Browse (which is selected), Search, Docs, News, Stats, About, and Say Thanks. At the bottom, there's a "Readme" link and a "SOURCE" link to [raw.githubusercontent.com](https://raw.githubusercontent.com/davidhq/SublimeText-Ethereum/master/). A large callout box highlights the "Ethereum Package for Sublime Text 2/3" and its "Description".

6. Install Sublime Text console http://docs.sublimetext.info/en/latest/getting_started/install.html <http://www.anujvarma.com/package-control-in-sublime-text-setting-up-for-solidity-development/>

The screenshot shows a web browser window with multiple tabs open. The active tab displays the 'Installation' section of the Sublime Text Unofficial Documentation. The page header includes the Sublime Text logo and the text 'Sublime Text Unofficial Documentation'. A search bar and a 'About This Documentation' link are also present. The main content area is titled 'Installation' and contains a note about reading the conditions for use. It then discusses the process of installing Sublime Text for different platforms: OS X, Windows, and Linux. Each platform section includes a sub-section for '32 bits or 64 bits?'. The OS X section notes that there is only one version for OS X. The Windows section suggests running the 64-bit version on modern Windows and the 32-bit version if having trouble. The Linux section provides a command ('uname -m') to check the operating system's type. A sidebar on the left lists various documentation categories like 'Basic Concepts', 'Editing', and 'Customization'. At the bottom of the sidebar is an advertisement placeholder for 'YOUR AD HERE' with the text 'Reach over 7 million devs each month when you [advertise with Read the Docs](#)'. The browser's address bar shows the URL: docs.sublimetext.info/en/latest/getting_started/install.html. The toolbar at the top includes standard browser icons like back, forward, and search.

Session 2: Project Setup & Smoke Test

1. Open GANACHE

Ganache					
ACCOUNTS	BLOCKS	TRANSACTIONS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES	⚙️
CURRENT BLOCK 0	GAS PRICE 200000000	GAS LIMIT 6721975	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING
MNEMONIC ⓘ record tower kangaroo wheat merit crawl almost auction chief poverty ripple fall					
ADDRESS 0xb6D6510c99C4387c6d51C3D36642b8CE5269f518	BALANCE 100.00 ETH		TX COUNT 0	INDEX 0	🔑
ADDRESS 0x0f453b50BecCdd51daD5966d07132e53d20D25Dd	BALANCE 100.00 ETH		TX COUNT 0	INDEX 1	🔑
ADDRESS 0x68b090fCB1349f24041b5D68Dae173936f418251	BALANCE 100.00 ETH		TX COUNT 0	INDEX 2	🔑
ADDRESS 0x7E4BE9B7d977f6717AD8d953933F0806B1cFee58	BALANCE 100.00 ETH		TX COUNT 0	INDEX 3	🔑
ADDRESS 0xd56ce63ce7FE32DE23A2E472C27152d542acA558	BALANCE 100.00 ETH		TX COUNT 0	INDEX 4	🔑
ADDRESS 0x5695118D159c742D5aeFbCba546FFd1801Bcd945	BALANCE 100.00 ETH		TX COUNT 0	INDEX 5	🔑
ADDRESS 0x2B47B32E48067ab399055CD0B0460Fa5124f3f69	BALANCE 100.00 ETH		TX COUNT 0	INDEX 6	🔑

2. Open Console: mkdir token_sale

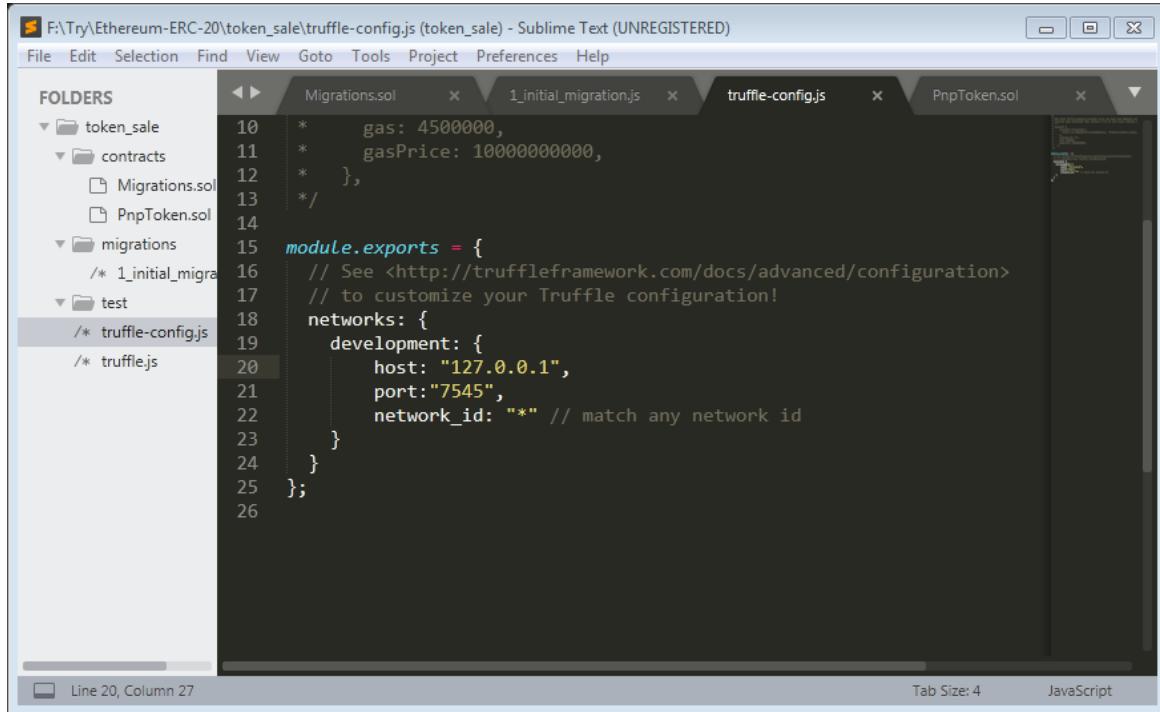
3. Go to token_sale> truffle init

```
Administrator: Windows Command Processor
unbox` must be executed in an empty folder. Stopping to prevent overwriting data.
    at C:\Users\Administrator\AppData\Roaming\npm\node_modules\truffle\build\webpack:\packages\truffle-box\lib\utils\unbox.js:22:1
    at <anonymous>
    at process._tickCallback (internal/process/next_tick.js:189:7)
    at Function.Module.runMain (module.js:696:11)
    at startup (bootstrap_node.js:204:16)
    at bootstrap_node.js:625:3

F:\Try\Ethereum-ERC-20>cd token_sale
F:\Try\Ethereum-ERC-20\token_sale>truffle init
Downloading...
Unpacking...
Setting up...
Unbox successful. Sweet!
Commands:
  Compile:      truffle compile
  Migrate:      truffle migrate
  Test contracts: truffle test
F:\Try\Ethereum-ERC-20\token_sale>subl .
```

4. Open Sublime Text: File> Open Folder token_sale

5. Go to truffle-config.js (for Windows) & add the following code



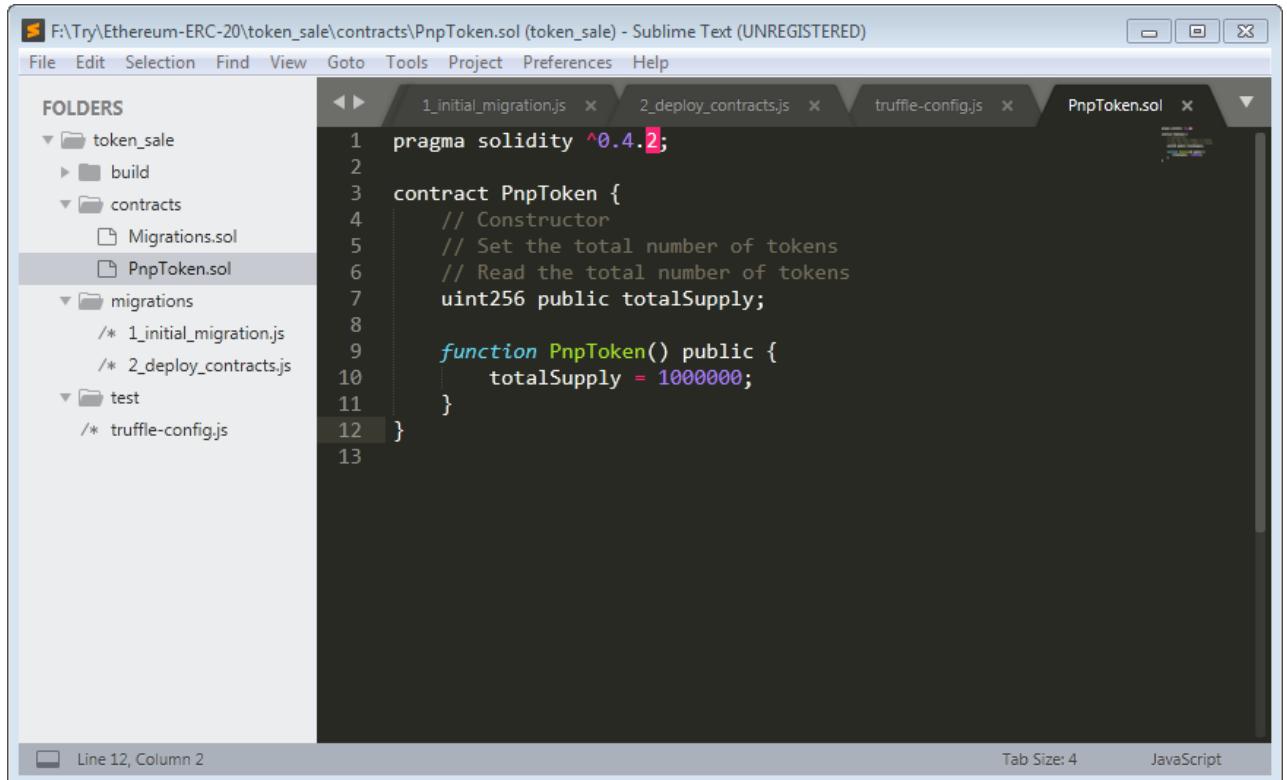
```

F:\Try\Ethereum-ERC-20\token_sale\truffle-config.js (token_sale) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
token_sale
  contracts
    Migrations.sol
    PnpToken.sol
  migrations
    /* 1_initial_migration.js
  test
  /* truffle-config.js
/* truffle.js
module.exports = {
  // See <http://truffleframework.com/docs/advanced/configuration>
  // to customize your Truffle configuration!
  networks: {
    development: {
      host: "127.0.0.1",
      port: "7545",
      network_id: "*" // match any network id
    }
  }
};

Line 20, Column 27
Tab Size: 4
JavaScript

```

6. Go to Folder token_sale\contracts create file “PnpToken.sol” & add the following code



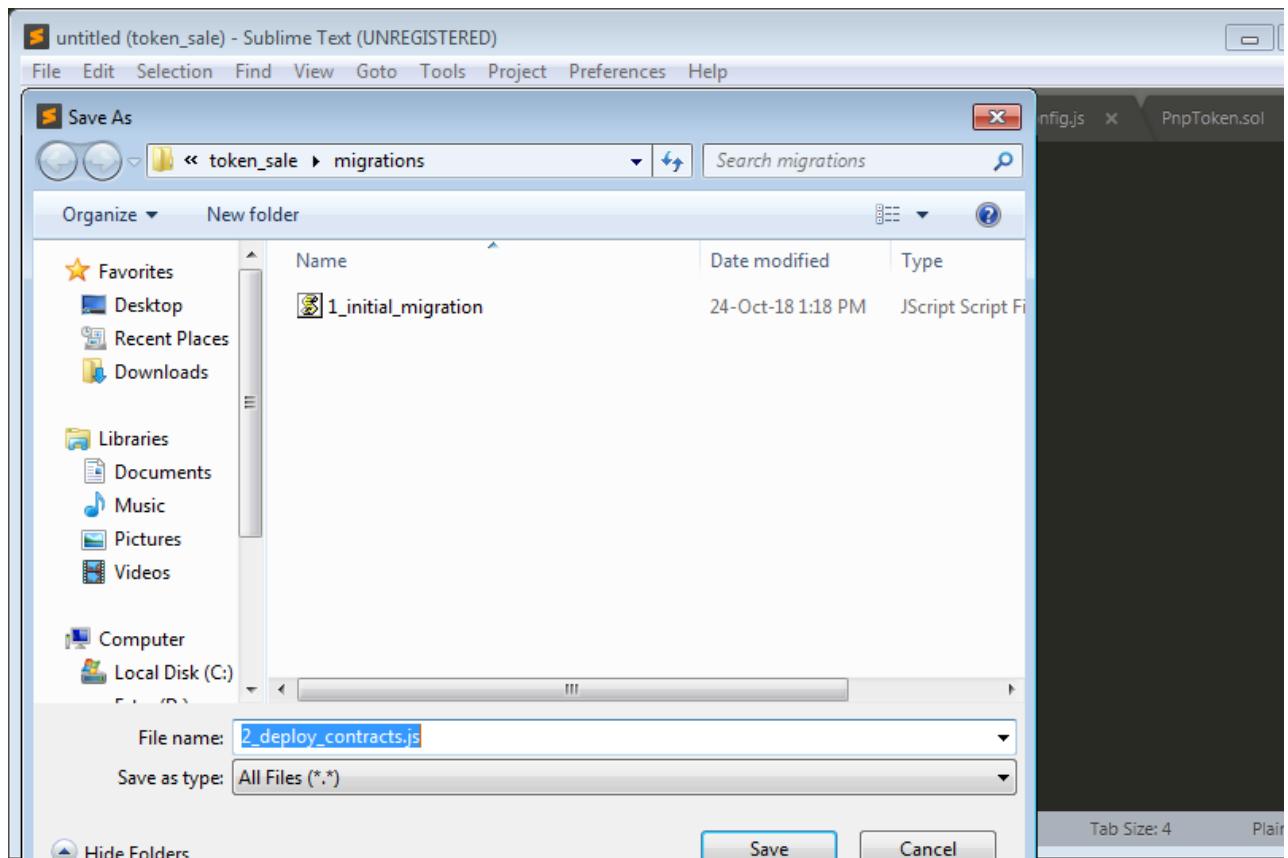
```

F:\Try\Ethereum-ERC-20\token_sale\contracts\PnpToken.sol (token_sale) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
token_sale
  build
  contracts
    Migrations.sol
    PnpToken.sol
  migrations
    /* 1_initial_migration.js
    /* 2_deploy_contracts.js
  test
  /* truffle-config.js
pragma solidity ^0.4.2;
contract PnpToken {
  // Constructor
  // Set the total number of tokens
  // Read the total number of tokens
  uint256 public totalSupply;
  function PnpToken() public {
    totalSupply = 1000000;
  }
}

Line 12, Column 2
Tab Size: 4
JavaScript

```

7. Create new file “2_deploy_contracts.js”



8. Open file “2_deploy_contracts.js” & add following code

```
var PnpToken = artifacts.require("./PnpToken.sol");
module.exports = function(deployer) {
  deployer.deploy(PnpToken);
};
```

9. Go to console type truffle migrate

```
Administrator: Windows Command Processor
'sUBL' is not recognized as an internal or external command,
operable program or batch file.

F:\Try\Ethereum-ERC-20\token_sale>touch contracts/PnpToken.sol
'touch' is not recognized as an internal or external command,
operable program or batch file.

F:\Try\Ethereum-ERC-20\token_sale>truffle migrate
F:\Try\Ethereum-ERC-20\token_sale>truffle
F:\Try\Ethereum-ERC-20\token_sale>truffle-config migrate
F:\Try\Ethereum-ERC-20\token_sale>truffle migrate
F:\Try\Ethereum-ERC-20\token_sale>truffle init
F:\Try\Ethereum-ERC-20\token_sale>truffle init
F:\Try\Ethereum-ERC-20\token_sale>truffle migrate
F:\Try\Ethereum-ERC-20\token_sale>truffle migrate
F:\Try\Ethereum-ERC-20\token_sale>truffle migrate
```

Note: If you are getting an error then remove truffle.js (For Windows Only)

```
Administrator: Windows Command Processor
Compilation failed. See above.

F:\Try\Ethereum-ERC-20\token_sale>truffle migrate
Compiling .\contracts\Migrations.sol...
Compiling .\contracts\PnpToken.sol...

Compilation warnings encountered:
/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:9:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){ ... }" instead.
    function PnpToken() public {
^ (Relevant source part starts here and spans across multiple lines).

Writing artifacts to .\build\contracts

Using network 'development'.

Running migration: 1_initial_migration.js
  Deploying Migrations...
  ... 0x33824209d299a0833436f36fefe446312f13db7ed5e8042d2ac40b4b931373df
  Migrations: 0x6382ff1efbf84a71045686018876fb73883190e0
Saving successful migration to network...
  ... 0xb7d0f7b0fc4c0742b0d1ebccdd2efbc1e318fdc112b44802db520392b663ddb3
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Deploying PnpToken...
  ... 0x308877d974a9647957a7a45eb297bc25f460fc0661f3bd78121dd7ba3614e73f
  PnpToken: 0xc02e669ac99404854b9750c0c4ce92ef2de0e68d
Saving successful migration to network...
  ... 0x35d419fd693b002448d46fc61ac3f7dda8d67597d88a86669ebbbb68b3ad8b96
Saving artifacts...

F:\Try\Ethereum-ERC-20\token_sale>
```

10. Perform following commands to verify our contract is working or not!!

```
F:\Try\Ethereum-ERC-20\token_sale>truffle console
```

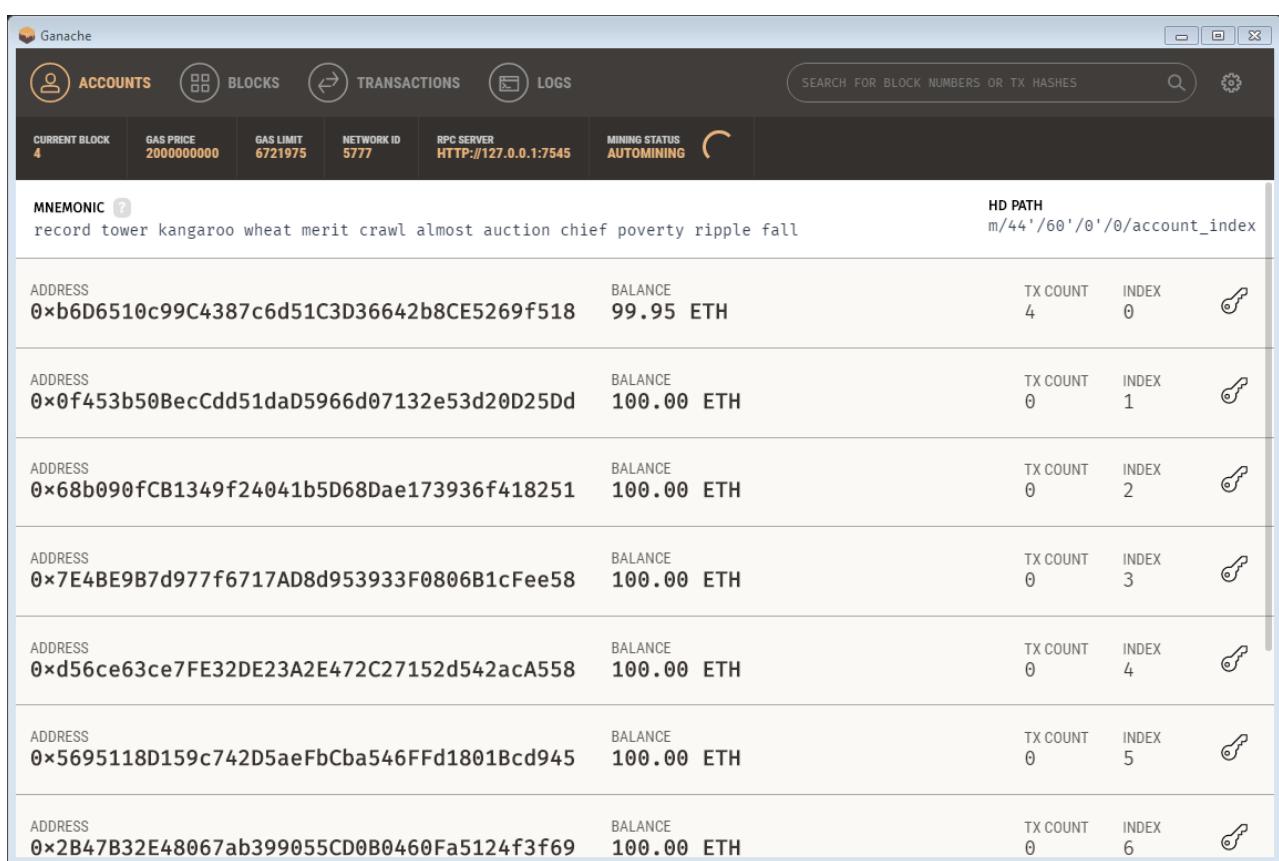
```
truffle(development)> PnpToken.deployed().then(function(i) {token=i;})
undefined
truffle(development)>
```

```
truffle(development)> token.address
'0xc02e669ac99404854b9750c0c4ce92ef2de0e68d'
truffle(development)>
```

```
truffle(development)> token.totalSupply().then(function(s) {totalSupply=s;})
undefined
truffle(development)> totalSupply
BigNumber { s: 1, e: 6, c: [ 1000000 ] }
truffle(development)> totalSupply.toNumber()
1000000
truffle(development)>
```

```
truffle(development)> .exit
```

```
F:\Try\Ethereum-ERC-20\token_sale>
```



The screenshot shows the Ganache interface with the following details:

Ganache					
ACCOUNTS	BLOCKS	TRANSACTIONS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES	⚙️
CURRENT BLOCK 4	GAS PRICE 2000000000	GAS LIMIT 6721975	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING
MNEMONIC record tower kangaroo wheat merit crawl almost auction chief poverty ripple fall					
HD PATH <code>m/44'/60'/0'/0/account_index</code>					
ADDRESS <code>0xb6D6510c99C4387c6d51C3D36642b8CE5269f518</code>	BALANCE 99.95 ETH	TX COUNT 4	INDEX 0		
ADDRESS <code>0x0f453b50BecCdd51daD5966d07132e53d20D25Dd</code>	BALANCE 100.00 ETH	TX COUNT 0	INDEX 1		
ADDRESS <code>0x68b090fCB1349f24041b5D68Dae173936f418251</code>	BALANCE 100.00 ETH	TX COUNT 0	INDEX 2		
ADDRESS <code>0x7E4BE9B7d977f6717AD8d953933F0806B1cFee58</code>	BALANCE 100.00 ETH	TX COUNT 0	INDEX 3		
ADDRESS <code>0xd56ce63ce7FE32DE23A2E472C27152d542acA558</code>	BALANCE 100.00 ETH	TX COUNT 0	INDEX 4		
ADDRESS <code>0x5695118D159c742D5aeFbCba546FFd1801Bcd945</code>	BALANCE 100.00 ETH	TX COUNT 0	INDEX 5		
ADDRESS <code>0x2B47B32E48067ab399055CD0B0460Fa5124f3f69</code>	BALANCE 100.00 ETH	TX COUNT 0	INDEX 6		

11. For Testing Smart Contract create a file “PnpToken.js”

The screenshot shows a Sublime Text window with the following details:

- File Path:** F:\Try\Ethereum-ERC-20\token_sale\test\PnpToken.js (token_sale) - Sublime Text (UNREGISTERED)
- File Tabs:** 1_initial_migration.js, 2_deploy_contracts.js, truffle-config.js, PnpToken.js
- Folders:** token_sale (containing build, contracts, migrations, test), contracts (containing Migrations.sol, PnpToken.sol), migrations (containing 1_initial_migration.js, 2_deploy_contracts.js), test (containing PnpToken.js, truffle-config.js).
- Status Bar:** Line 1, Column 1, Tab Size: 4, JavaScript.

The screenshot shows the content of the PnpToken.js file in Sublime Text:

```
1 var PnpToken = artifacts.require("./PnpToken");
2
3 contract('PnpToken', function (accounts) {
4     // body...
5     it('sets the total supply upon deployment', function() {
6         return PnpToken.deployed().then(function(instance){
7             tokenInstance= instance;
8             return tokenInstance.totalSupply();
9         }).then(function(totalSupply){
10             assert.equal(totalSupply.toNumber(),1000000,'sets th
11         }));
12     });
13 })
```

The code defines a Truffle test for the PnpToken smart contract. It uses the `artifacts.require` method to load the artifact for the PnpToken contract. The test then deploys the contract and checks its total supply upon deployment, asserting that it is 1,000,000.

File Path: F:\Try\Ethereum-ERC-20\token_sale\test\PnpToken.js (token_sale) - Sublime Text (UNREGISTERED)

File Tabs: 1_initial_migration.js, 2_deploy_contracts.js, truffle-config.js, PnpToken.js

Folders: token_sale (containing build, contracts, migrations, test), contracts (containing Migrations.sol, PnpToken.sol), migrations (containing 1_initial_migration.js, 2_deploy_contracts.js), test (containing PnpToken.js, truffle-config.js).

Status Bar: Line 13, Column 4, Tab Size: 4, JavaScript.

```
Administrator: Windows Command Processor
'truffle' is not recognized as an internal or external command,
operable program or batch file.

F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Contract: PnpToken
  ✓ sets the total supply upon deployment

  1 passing (50ms)

E:\Try\Ethereum-ERC-20\token_sale>
```

The screenshot shows a Sublime Text window with the following details:

- File Path:** F:\Try\Ethereum-ERC-20\token_sale\test\PnpToken.js
- File Type:** (token_sale) - Sublime Text (UNREGISTERED)
- Code Content:**

```
var PnpToken = artifacts.require("./PnpToken");
contract('PnpToken', function (accounts) {
  // body...
  it('sets the total supply upon deployment', function() {
    return PnpToken.deployed().then(function(instance){
      tokenInstance= instance;
      return tokenInstance.totalSupply();
    }).then(function(totalSupply){
      assert.equal(totalSupply.toNumber(),9000000, 'sets th
    }));
  });
})
```
- Folders:** token_sale, build, contracts (containing Migrations.sol and PnpToken.sol), migrations (containing 1_initial_migration.js and 2_deploy_contracts.js), and test (containing PnpToken.js and truffle-config.js).
- Status Bar:** 7 characters selected, Tab Size: 4, JavaScript.

```
Administrator: Windows Command Processor
    ✓ sets the total supply upon deployment

    1 passing (44ms)

F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Contract: PnpToken
  1) sets the total supply upon deployment
    > No events were emitted

    0 passing (55ms)
    1 failing

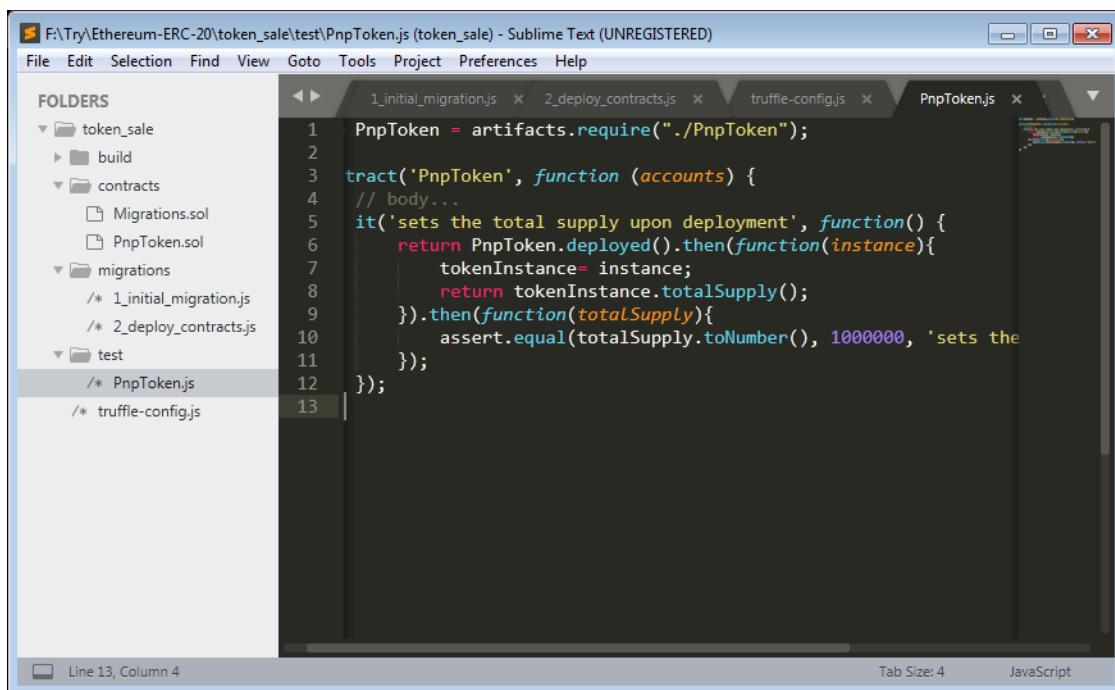
  1) Contract: PnpToken
    sets the total supply upon deployment:

      sets the total supply to 10,00,000
      + expected - actual

      -1000000
      +9000000

      at test\PnpToken.js:10:11
      at <anonymous>
      at process._tickCallback (internal/process/next_tick.js:189:7)

F:\Try\Ethereum-ERC-20\token_sale>
```



The screenshot shows the Sublime Text editor with the following details:

- File Path:** F:\Try\Ethereum-ERC-20\token_sale\test\PnpToken.js (token_sale) - Sublime Text (UNREGISTERED)
- File Tabs:** 1_initial_migration.js, 2_deploy_contracts.js, truffle-config.js, PnpToken.js
- Folders:** token_sale, build, contracts (Migrations.sol, PnpToken.sol), migrations (1_initial_migration.js, 2_deploy_contracts.js), test (PnpToken.js, truffle-config.js)
- PnpToken.js Content:**

```

1 PnpToken = artifacts.require("./PnpToken");
2
3 contract('PnpToken', function (accounts) {
4   // body...
5   it('sets the total supply upon deployment', function() {
6     return PnpToken.deployed().then(function(instance){
7       tokenInstance= instance;
8       return tokenInstance.totalSupply();
9     }).then(function(totalSupply){
10       assert.equal(totalSupply.toNumber(), 1000000, 'sets the
11     });
12   });
13 });

```
- Status Bar:** Line 13, Column 4, Tab Size: 4, JavaScript

```
Administrator: Windows Command Processor
1) sets the total supply upon deployment
> No events were emitted

0 passing (55ms)
1 failing

1) Contract: PnpToken
   sets the total supply upon deployment:
     sets the total supply to 10,00,000
     + expected - actual
     -1000000
     +9000000
   at test\PnpToken.js:10:11
   at <anonymous>
   at process._tickCallback (internal/process/next_tick.js:189:7)

F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Contract: PnpToken
  ✓ sets the total supply upon deployment

1 passing (57ms)

F:\Try\Ethereum-ERC-20\token_sale>truffle test
```

Session 3: Creating the ERC-20 Token Smart Contract

1. Open sublime and modify code as below (Observe PnpToken function)

The screenshot shows a Sublime Text window with the file `PnpToken.sol` open. The code defines a contract named `PnpToken` with a constructor that sets the total supply to `_initialSupply`. The code is as follows:

```
1 pragma solidity ^0.4.2;
2
3 contract PnpToken {
4     // Constructor
5     // Set the total number of tokens
6     // Read the total number of tokens
7     uint256 public totalSupply;
8
9     function PnpToken(uint256 _initialSupply) public {
10         totalSupply = _initialSupply;
11     }
12 }
13
```

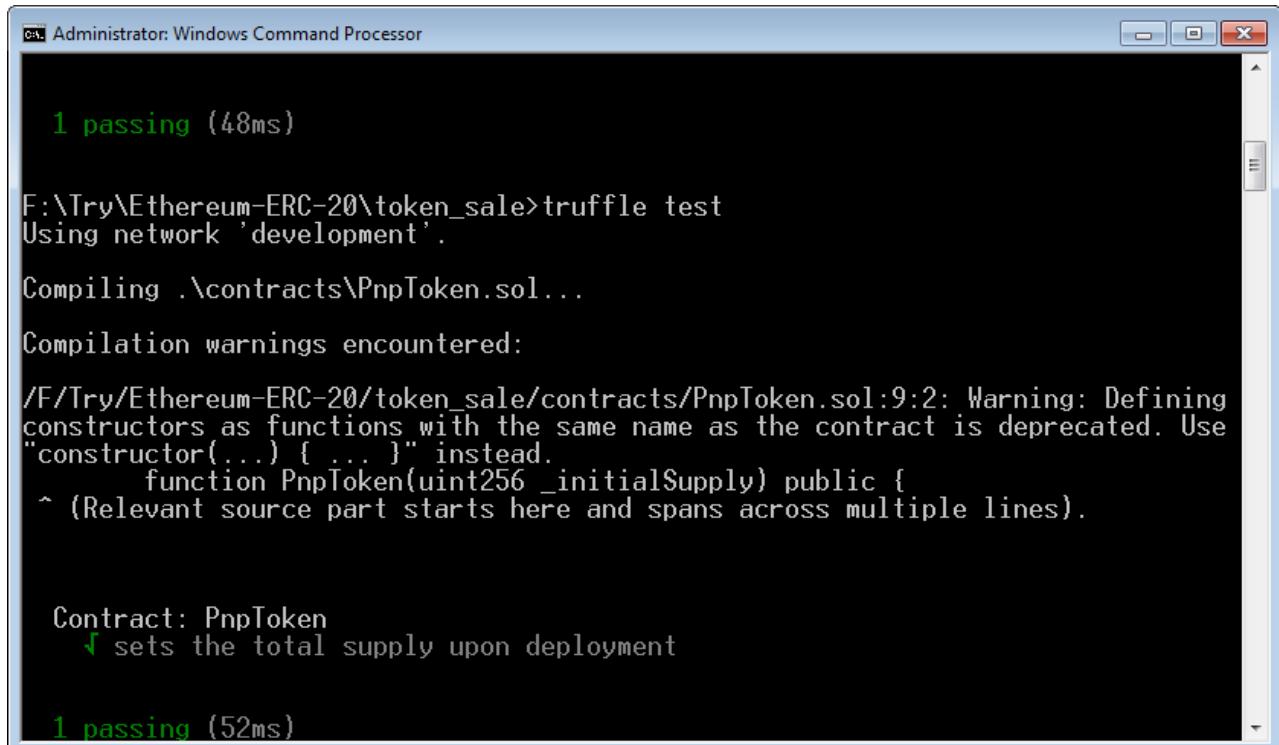
The Sublime Text interface includes a sidebar with project files like `token_sale`, `build`, `contracts` (containing `Migrations.sol` and `PnpToken.sol`), `migrations` (containing `1_initial_migration.js` and `2_deploy_contracts.js`), and `test` (containing `PnpToken.js` and `truffle-config.js`). The status bar at the bottom shows "Line 13, Column 1", "Tab Size: 4", and "JavaScript".

The screenshot shows a Sublime Text window with the file `2_deploy_contracts.js` open. The code uses Truffle artifacts to deploy the `PnpToken` contract with an initial supply of 1000000. The code is as follows:

```
1 var PnpToken = artifacts.require("./PnpToken.sol");
2
3 module.exports = function(deployer) {
4     deployer.deploy(PnpToken, 1000000);
5 };
6
```

The Sublime Text interface includes a sidebar with project files like `token_sale`, `build`, `contracts` (containing `Migrations.sol` and `PnpToken.sol`), `migrations` (containing `1_initial_migration.js` and `2_deploy_contracts.js`), and `test` (containing `PnpToken.js` and `truffle-config.js`). The status bar at the bottom shows "Line 6, Column 1", "Tab Size: 4", and "JavaScript".

2. Go to console and verify the updated contract



```
Administrator: Windows Command Processor
1 passing (48ms)

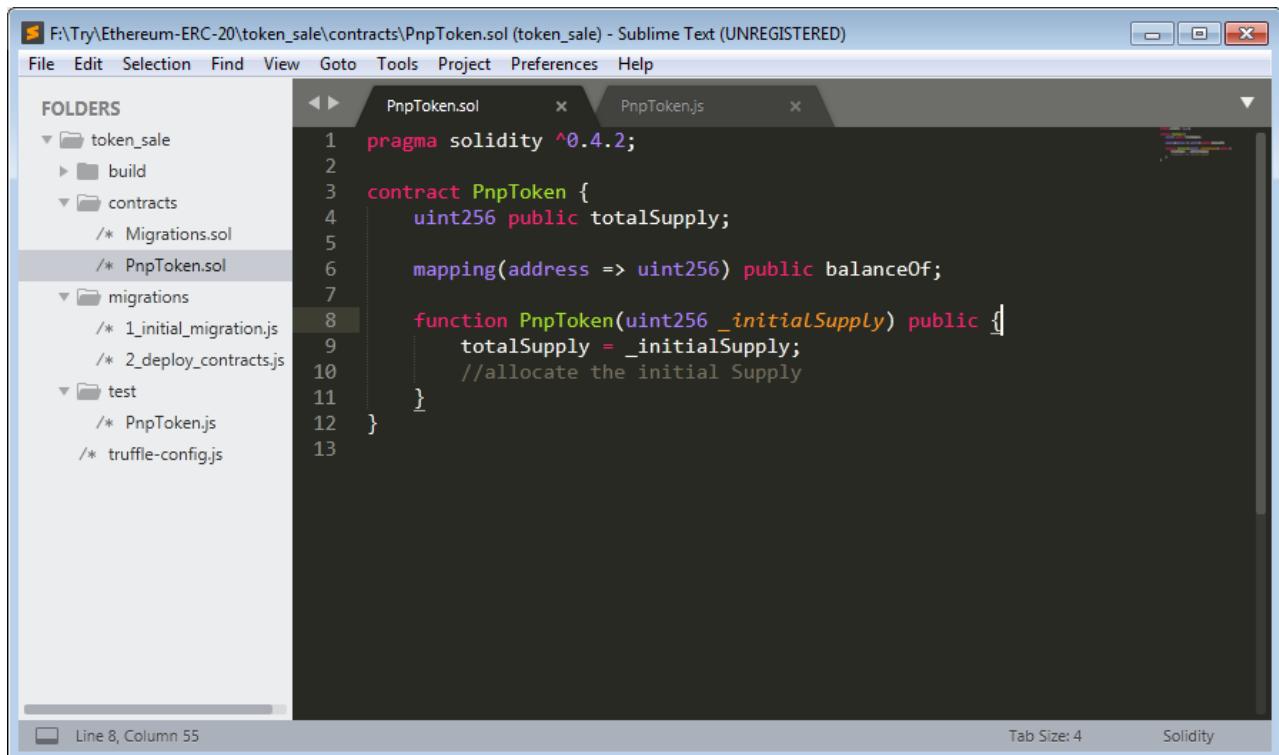
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:
/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:9:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){ ... }" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).

Contract: PnpToken
  ✓ sets the total supply upon deployment

1 passing (52ms)
```

3. Verify PnpToken for different address



```
F:\Try\Ethereum-ERC-20\token_sale\contracts\PnpToken.sol (token_sale) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
token_sale
  build
  contracts
    /* Migrations.sol
    /* PnpToken.sol
  migrations
    /* 1_initial_migration.js
    /* 2_deploy_contracts.js
  test
    /* PnpToken.js
    /* truffle-config.js

PnpToken.sol
1 pragma solidity ^0.4.2;
2
3 contract PnpToken {
4     uint256 public totalSupply;
5
6     mapping(address => uint256) public balanceOf;
7
8     function PnpToken(uint256 _initialSupply) public {
9         totalSupply = _initialSupply;
10        //allocate the initial Supply
11    }
12 }
13

Line 8, Column 55
Tab Size: 4
Solidity
```

```

F:\Try\Ethereum-ERC-20\token_sale\test\PnpToken.js (token_sale) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
token_sale
build
contracts
  / Migrations.sol
  / PnpToken.sol
migrations
  / 1_initial_migration.js
  / 2_deploy_contracts.js
test
  / PnpToken.js
  / truffle-config.js

PnpToken.sol x PnpToken.js x
1 var PnpToken = artifacts.require("./PnpToken.sol");
2
3 contract('PnpToken', function(accounts) {
4   var tokenInstance;
5
6     it('sets the total supply upon deployment', function() {
7       return PnpToken.deployed().then(function(instance){
8         tokenInstance = instance;
9         return tokenInstance.totalSupply();
10      }).then(function(totalSupply){
11        assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 10,00,000');
12        return tokenInstance.balanceOf(accounts[0]);
13      }).then(function(adminBalance){
14        assert.equal(adminBalance.toNumber(), 100000, 'it allocates the initial supply to admin account');
15      });
16    });
17  });

```

Line 17, Column 4 Tab Size: 4 JavaScript

```

Administrator: Windows Command Processor
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:

/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:8:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){ ... }" instead.
  function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).

Contract: PnpToken
1) sets the total supply upon deployment
  > No events were emitted

0 passing (91ms)
1 failing

1) Contract: PnpToken
  sets the total supply upon deployment:
    it allocates the initial supply to admin account
      + expected - actual
      -0
      +100000

      at test\PnpToken.js:14:11
      at <anonymous>
      at process._tickCallback (internal/process/next_tick.js:189:7)

```

Test will Fail

4. Now modify the code as below

```

F:\Try\Ethereum-ERC-20\token_sale\test\PnpToken.js (token_sale) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
token_sale
  build
  contracts
    /* Migrations.sol
     * PnpToken.sol
  migrations
    /* 1_initial_migration.js
    /* 2_deploy_contracts.js
  test
    /* PnpToken.js
     * truffle-config.js
PnPToken.sol      PnPToken.js
1 var PnPToken = artifacts.require("./PnPToken.sol");
2
3 contract('PnPToken', function (accounts) {
4   var tokenInstance;
5
6   it('sets the total supply upon deployment', function() {
7     return PnPToken.deployed().then(function(instance){
8       tokenInstance = instance;
9       return tokenInstance.totalSupply();
10    }).then(function(totalSupply){
11      assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 10,00,000');
12      return tokenInstance.balanceOf(accounts[0]);
13    }).then(function(adminBalance){
14      assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the initial supply to admin account')
15    });
16  });
17 });

```

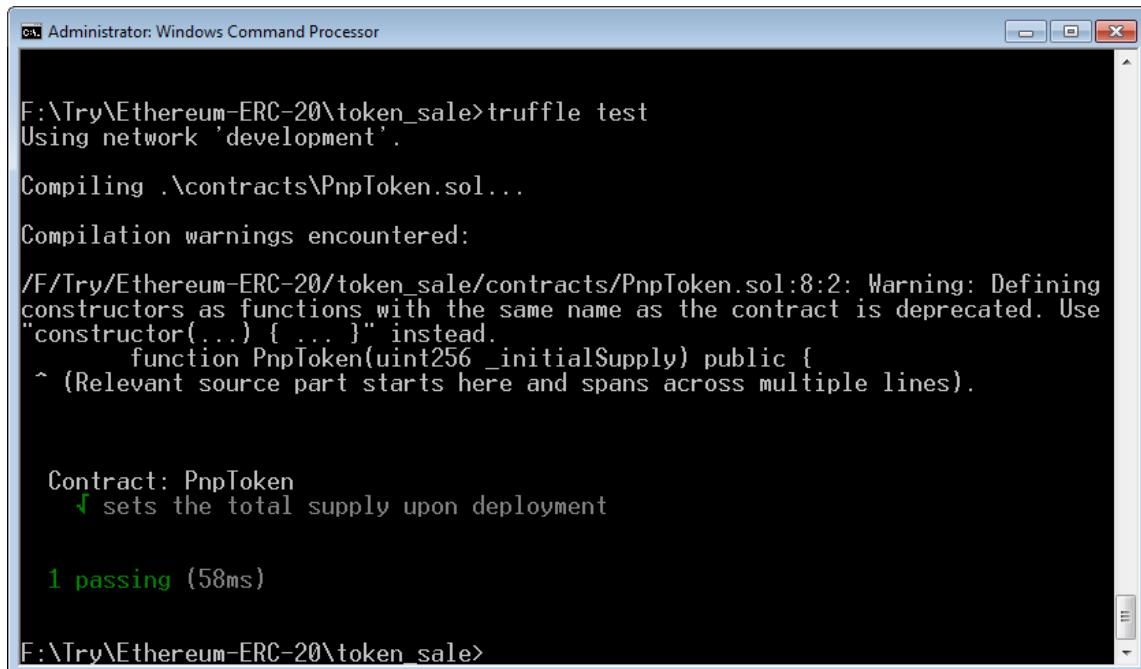
Line 16, Column 8 Tab Size: 4 JavaScript

```

F:\Try\Ethereum-ERC-20\token_sale\contracts\PnpToken.sol (token_sale) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
token_sale
  build
  contracts
    /* Migrations.sol
     * PnpToken.sol
  migrations
    /* 1_initial_migration.js
    /* 2_deploy_contracts.js
  test
    /* PnpToken.js
     * truffle-config.js
PnPToken.sol      PnPToken.js
1 pragma solidity ^0.4.2;
2
3 contract PnPToken {
4   uint256 public totalSupply;
5
6   mapping(address => uint256) public balanceOf;
7
8   function PnPToken(uint256 _initialSupply) public {
9     balanceOf[msg.sender] = _initialSupply; // Key Value
10    totalSupply = _initialSupply;
11  }
12
13

```

Line 13, Column 1 Tab Size: 4 Solidity



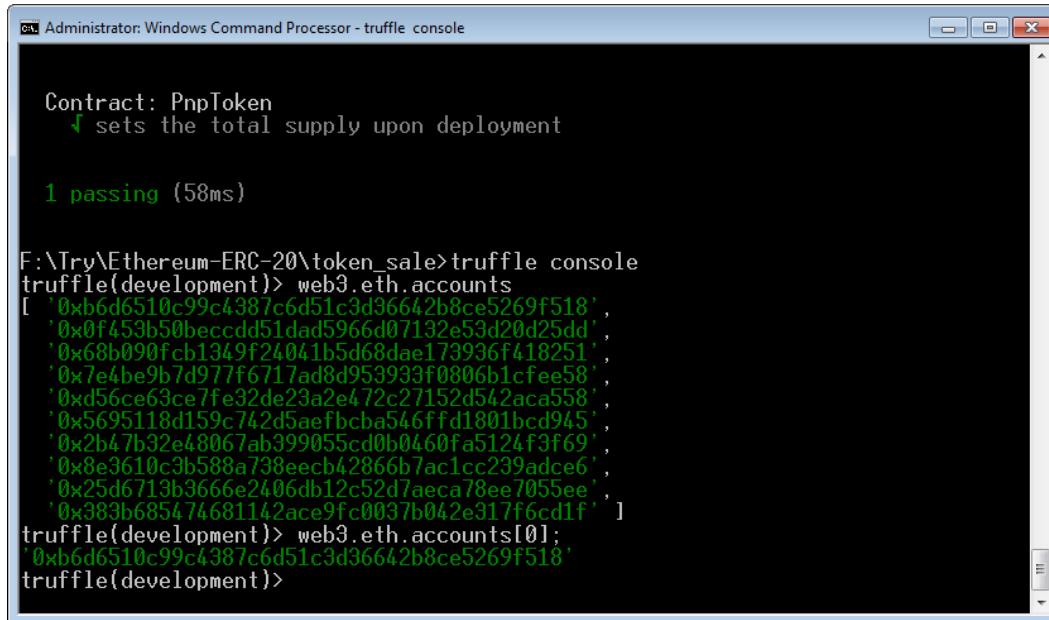
```
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.
Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:
/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:8:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){ ... }" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).

Contract: PnpToken
  ✓ sets the total supply upon deployment

1 passing (58ms)

F:\Try\Ethereum-ERC-20\token_sale>
```

5. Find out accounts via console & verify



```
Administrator: Windows Command Processor - truffle console
Contract: PnpToken
  ✓ sets the total supply upon deployment

1 passing (58ms)

F:\Try\Ethereum-ERC-20\token_sale>truffle console
truffle(development)> web3.eth.accounts
[ '0xb6d6510c99c4387c6d51c3d36642b8ce5269f518',
  '0x0f459b50beccdd51dad5966d07132e53d20d25dd',
  '0x68b090fcb1349f24041b5d68dae173936f418251',
  '0x7e4be9b7d977f6717ad8d953933f0806b1cfee58',
  '0xd56ce63ce7fe32de23a2e472c27152d542aca558',
  '0x5695118d159c742d5aefbcba546ffd1801bcd945',
  '0x2b47b32e48067ab399055cd0b0460fa5124f3f69',
  '0x8e3610c3b588a738eecb42866b7ac1cc239adce6',
  '0x25d6713b3666e2406db12c52d7aec78ee7055ee',
  '0x383b685474681142ace9fc0037b042e317f6cd1f' ]
truffle(development)> web3.eth.accounts[0];
'0xb6d6510c99c4387c6d51c3d36642b8ce5269f518'
truffle(development)>
```

Ganache					SEARCH FOR BLOCK NUMBERS OR TX HASHES	LOGS
ACCOUNTS	BLOCKS	TRANSACTIONS	LOGS			
CURRENT BLOCK 52	GAS PRICE 2000000000	GAS LIMIT 6721975	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING	()
MNEMONIC ? record tower kangaroo wheat merit crawl almost auction chief poverty ripple fall					HD PATH m/44'/60'/0'/0/account_index	
ADDRESS 0xb6D6510c99C4387c6d51C3D36642b8CE5269f518	BALANCE 99.35 ETH			TX COUNT 52	INDEX 0	🔗
ADDRESS 0x0f453b50BecCdd51daD5966d07132e53d20D25Dd	BALANCE 100.00 ETH			TX COUNT 0	INDEX 1	🔗
ADDRESS 0x68b090fcB1349f24041b5D68Dae173936f418251	BALANCE 100.00 ETH			TX COUNT 0	INDEX 2	🔗
ADDRESS 0x7E4BE9B7d977f6717AD8d953933F0806B1cFee58	BALANCE 100.00 ETH			TX COUNT 0	INDEX 3	🔗
ADDRESS 0xd56ce63ce7FE32DE23A2E472C27152d542acA558	BALANCE 100.00 ETH			TX COUNT 0	INDEX 4	🔗
ADDRESS 0x5695118D159c742D5aeFbCba546FFd1801Bcd945	BALANCE 100.00 ETH			TX COUNT 0	INDEX 5	🔗
ADDRESS 0x2B47B32E48067ab399055CD0B0460Fa5124f3f69	BALANCE 100.00 ETH			TX COUNT 0	INDEX 6	🔗

6. Now give the name, symbol & standard to token

F:\Try\Ethereum-ERC-20\token_sale\contracts\PnpToken.sol (token_sale) - Sublime Text (UNREGISTERED)

```

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
token_sale
  build
  contracts
    Migrations.sol
    PnpToken.sol
  migrations
    1_initial_migration.js
    2_deploy_contracts.js
  test
    PnpToken.js
    truffle-config.js

PnpToken.sol
pragma solidity ^0.4.2;

contract PnpToken {
  //Name
  string public name="PNP Token";
  // Symbol
  uint256 public totalSupply;
  mapping(address => uint256) public balanceOf;

  function PnpToken(uint256 _initialSupply) public {
    balanceOf[msg.sender] = _initialSupply; // Key Value
    totalSupply = _initialSupply;
  }
}

Line 16, Column 1
Tab Size: 4
Solidity

```

The screenshot shows the Sublime Text interface with two tabs open: 'PnpToken.sol' and 'PnpToken.js'. The left sidebar displays the project structure:

```

F:\Try\Ethereum-ERC-20\token_sale\test\PnpToken.js (token_sale) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
token_sale
  build
  contracts
    Migrations.sol
    PnpToken.sol
  migrations
    1_initial_migration.js
    2_deploy_contracts.js
  test
    PnpToken.js
  truffle-config.js

```

The 'PnpToken.js' file contains the following Truffle test code:

```

var PnpToken = artifacts.require("./PnpToken.sol");

contract('PnpToken', function (accounts) {
  var tokenInstance;

  it('initializes the contract with correct values', function() {
    return PnpToken.deployed().then(function(instance){
      tokenInstance = instance;
      return tokenInstance.name();
    }).then(function(name){
      assert.equal(name,'PNP Token','has the correct name');
    });
  });

  it('allocates the initial supply upon deployment', function() {
    return PnpToken.deployed().then(function(instance){
      tokenInstance = instance;
      return tokenInstance.totalSupply();
    }).then(function(totalSupply){
      assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply');
      return tokenInstance.balanceOf(accounts[0]);
    }).then(function(adminBalance){
      assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the correct amount');
    });
  });
});

```

Line 10 is highlighted.

The screenshot shows a Windows Command Processor window titled 'Administrator: Windows Command Processor' with the following output:

```

F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:

/F\F:\Try\Ethereum-ERC-20\token_sale\contracts\PnpToken.sol:11:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){ ... }" instead.
  function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).

Contract: PnpToken
  ✓ initializes the contract with correct values
  ✓ allocates the initial supply upon deployment

2 passing (91ms)

F:\Try\Ethereum-ERC-20\token_sale>

```

F:\Try\Ethereum-ERC-20\token_sale\contracts\PnpToken.sol (token_sale) - Sublime Text (UNREGISTERED)

```

File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
token_sale
  build
  contracts
    /* Migrations.sol
    /* PnpToken.sol
      migrations
        /* 1_initial_migration.js
        /* 2_deploy_contracts.js
    test
      /* PnpToken.js
      /* truffle-config.js
PnPToken.sol x PnPToken.js x
1 pragma solidity ^0.4.2;
2
3 contract PnPToken {
4
5     string public name="PNP Token"; //Name
6     string public symbol="PNP"; // Symbol
7
8     uint256 public totalSupply;
9
10    mapping(address => uint256) public balanceOf;
11
12    function PnPToken(uint256 _initialSupply) public {
13        balanceOf[msg.sender] = _initialSupply; // Key Value
14        totalSupply = _initialSupply;
15    }
16
17 }
```

Line 17, Column 1; Saved F:\Try\Ethereum-ERC-20\token_sale\contracts\PnpToken.sol (UTF-8) Tab Size: 4 Solidity

F:\Try\Ethereum-ERC-20\token_sale\test\PnpToken.js (token_sale) - Sublime Text (UNREGISTERED)

```

File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
token_sale
  build
  contracts
    /* Migrations.sol
    /* PnpToken.sol
      migrations
        /* 1_initial_migration.js
        /* 2_deploy_contracts.js
    test
      /* PnpToken.js
      /* truffle-config.js
PnPToken.sol x PnPToken.js x
1 var PnPToken = artifacts.require("./PnPToken.sol");
2
3 contract('PnPToken', function (accounts) {
4     var tokenInstance;
5
6     it('initializes the contract with correct values', function() {
7         return PnPToken.deployed().then(function(instance){
8             tokenInstance = instance;
9             return tokenInstance.name();
10            }).then(function(name){
11                assert.equal(name,'PNP Token','has the correct name');
12                return tokenInstance.symbol();
13            }).then(function(symbol){
14                assert.equal(symbol,'PNP','has the correct symbol');
15            });
16
17
18    it('allocates the initial supply upon deployment', function() {
19        return PnPToken.deployed().then(function(instance){
20            tokenInstance = instance;
21            return tokenInstance.totalSupply();
22        }).then(function(totalSupply){
23            assert.equal(totalSupply.toNumber(), 1000000, 'sets the total sup');
24            return tokenInstance.balanceOf(accounts[0]);
25        }).then(function(adminBalance){
26            assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the');
27        });
28    });
29});
```

Line 16, Column 7 Tab Size: 4 JavaScript

```

Administrator: Windows Command Processor

F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...

Compilation warnings encountered:

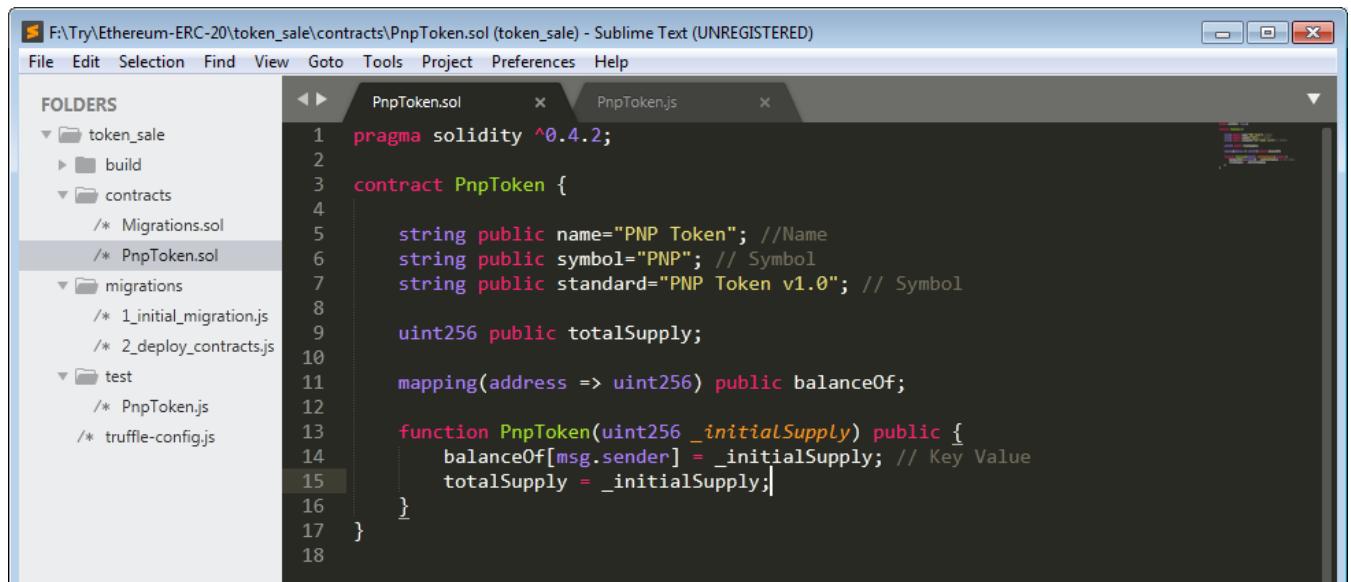
/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:12:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){...}" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).

Contract: PnpToken
  ✓ initializes the contract with correct values (79ms)
  ✓ allocates the initial supply upon deployment (42ms)

2 passing (161ms)

F:\Try\Ethereum-ERC-20\token_sale>

```



The screenshot shows the Sublime Text editor interface with the following details:

- File Path:** F:\Try\Ethereum-ERC-20\token_sale\contracts\PnpToken.sol (token_sale) - Sublime Text (UNREGISTERED)
- File Tabs:** PnpToken.sol (active), PnpToken.js
- Folders:** token_sale, build, contracts, migrations, test
- PnpToken.sol Content:**

```

1 pragma solidity ^0.4.2;
2
3 contract PnpToken {
4
5     string public name="PNP Token"; //Name
6     string public symbol="PNP"; // Symbol
7     string public standard="PNP Token v1.0"; // Symbol
8
9     uint256 public totalSupply;
10
11    mapping(address => uint256) public balanceOf;
12
13    function PnpToken(uint256 _initialSupply) public {
14        balanceOf[msg.sender] = _initialSupply; // Key Value
15        totalSupply = _initialSupply;
16    }
17}
18

```

```

F:\Try\Ethereum-ERC-20\token_sale\test\PnpToken.js (token_sale) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
token_sale
  build
  contracts
    /* Migrations.sol
     * PnpToken.sol
  migrations
    /* 1_initial_migration.js
     * 2_deploy_contracts.js
  test
    /* PnpToken.js
  truffle-config.js
PnPToken.sol x PnPToken.js x
1 var PnPToken = artifacts.require("./PnPToken.sol");
2
3 contract('PnPToken', function (accounts) {
4   var tokenInstance;
5
6   it('initializes the contract with correct values', function() {
7     return PnPToken.deployed().then(function(instance){
8       tokenInstance = instance;
9       return tokenInstance.name();
10      }).then(function(name){
11        assert.equal(name,'PNP Token','has the correct name');
12        return tokenInstance.symbol();
13      }).then(function(symbol){
14        assert.equal(symbol,'PNP','has the correct symbol');
15        return tokenInstance.standard();
16      }).then(function(stdandard){
17        assert.equal(stdandard,'PNP Token v1.0','has the correct standard');
18      });
19    });
20
21   it('allocates the initial supply upon deployment', function() {
22     return PnPToken.deployed().then(function(instance){
23       tokenInstance = instance;
24       return tokenInstance.totalSupply();
25     }).then(function(totalSupply){
26       assert.equal(totalSupply.toNumber(), 1000000, 'sets the total sup
27       return tokenInstance.balanceOf(accounts[0]);
28     }).then(function(adminBalance){
29       assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the
30     });
31   });
32 });

```

8 characters selected Tab Size: 4 JavaScript

```

Administrator: Windows Command Processor
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:

/F\Try\Ethereum-ERC-20\token_sale\contracts\PnpToken.sol:13:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){...}" instead.
  function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).

Contract: PnpToken
  ✓ initializes the contract with correct values (58ms)
  ✓ allocates the initial supply upon deployment

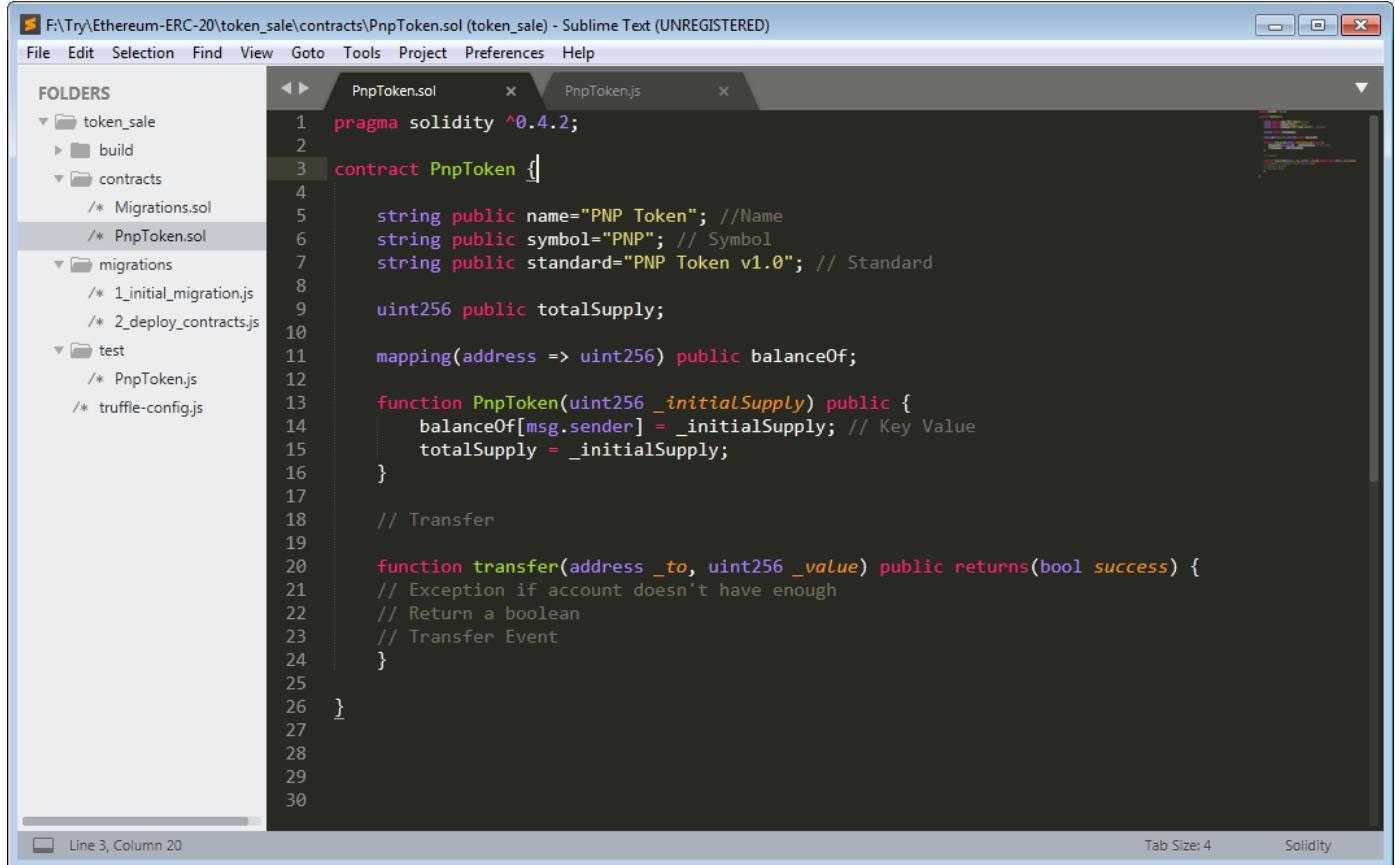
2 passing (152ms)

F:\Try\Ethereum-ERC-20\token_sale>

```

Session 4: Transferring Tokens

1. Add transfer token function



The screenshot shows a Sublime Text window with two tabs: 'PnPToken.sol' and 'PnPToken.js'. The 'PnPToken.sol' tab contains the following Solidity code:

```
pragma solidity ^0.4.2;

contract PnPToken {
    string public name = "PNP Token"; //Name
    string public symbol = "PNP"; // Symbol
    string public standard = "PNP Token v1.0"; // Standard

    uint256 public totalSupply;

    mapping(address => uint256) public balanceOf;

    function PnPToken(uint256 _initialSupply) public {
        balanceOf[msg.sender] = _initialSupply; // Key Value
        totalSupply = _initialSupply;
    }

    // Transfer
    function transfer(address _to, uint256 _value) public returns(bool success) {
        // Exception if account doesn't have enough
        // Return a boolean
        // Transfer Event
    }
}
```

The code defines a contract named PnPToken with variables for name, symbol, and standard, and functions for totalSupply and transfer.

The screenshot shows a Sublime Text window with two tabs: 'PnpToken.sol' and 'PnpToken.js'. The 'PnpToken.js' tab is active, displaying a block of JavaScript code. The code is a series of 'it' blocks, each testing a different aspect of a deployed Ethereum contract. The contract is named 'PnpToken'. The tests include checking the name ('PNP Token'), symbol ('PNP'), standard ('PNP Token v1.0'), total supply (1,000,000), and admin balance (1,000,000). It also includes a test for transferring ownership, which fails with a revert error message.

```
var PnpToken = artifacts.require("./PnpToken.sol");
contract('PnpToken', function (accounts) {
    var tokenInstance;
    it('initializes the contract with correct values', function() {
        return PnpToken.deployed().then(function(instance){
            tokenInstance = instance;
            return tokenInstance.name();
        }).then(function(name){
            assert.equal(name,'PNP Token','has the correct name');
            return tokenInstance.symbol();
        }).then(function(symbol){
            assert.equal(symbol,'PNP','has the correct symbol');
            return tokenInstance.standard();
        }).then(function(standard){
            assert.equal(standard,'PNP Token v1.0','has the correct standard');
        });
    });
    it('allocates the initial supply upon deployment', function() {
        return PnpToken.deployed().then(function(instance){
            tokenInstance = instance;
            return tokenInstance.totalSupply();
        }).then(function(totalSupply){
            assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 10,00,000');
            return tokenInstance.balanceOf(accounts[0]);
        }).then(function(adminBalance){
            assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the initial supply to admin account');
        });
    });
    it('transfers token ownership', function() {
        return PnpToken.deployed().then(function(instance){
            tokenInstance = instance;
            // Test 'require' statement first by transferring something larger than the sender's balance
            return tokenInstance.transfer.call(accounts[1],777777777777777);
        }).then(assert.fail).catch(function(error){
            assert(error.message.indexOf('revert') >= 0,'error message must contain revert');
        });
    });
});
```

```
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:

/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:13:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){...}" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:20:20: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war
ning.
    function transfer(address _to, uint256 _value) public returns(bool succe
ss) {
^-----^
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:20:33: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war
ning.
    function transfer(address _to, uint256 _value) public returns(bool succe
ss) {
^-----^
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:20:64: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war
ning.
    function transfer(address _to, uint256 _value) public returns(bool succe
ss) {
^-----^
-
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:20:2: Warning: Function
state mutability can be restricted to pure
```

```
Administrator: Windows Command Processor
ss) {
^-----^
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:20:33: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war
ning.
    function transfer(address _to, uint256 _value) public returns(bool succe
ss) {
^-----^
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:20:64: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war
ning.
    function transfer(address _to, uint256 _value) public returns(bool succe
ss) {
^-----^
-
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:20:2: Warning: Function
state mutability can be restricted to pure
    function transfer(address _to, uint256 _value) public returns(bool succe
ss) {
^ (Relevant source part starts here and spans across multiple lines).

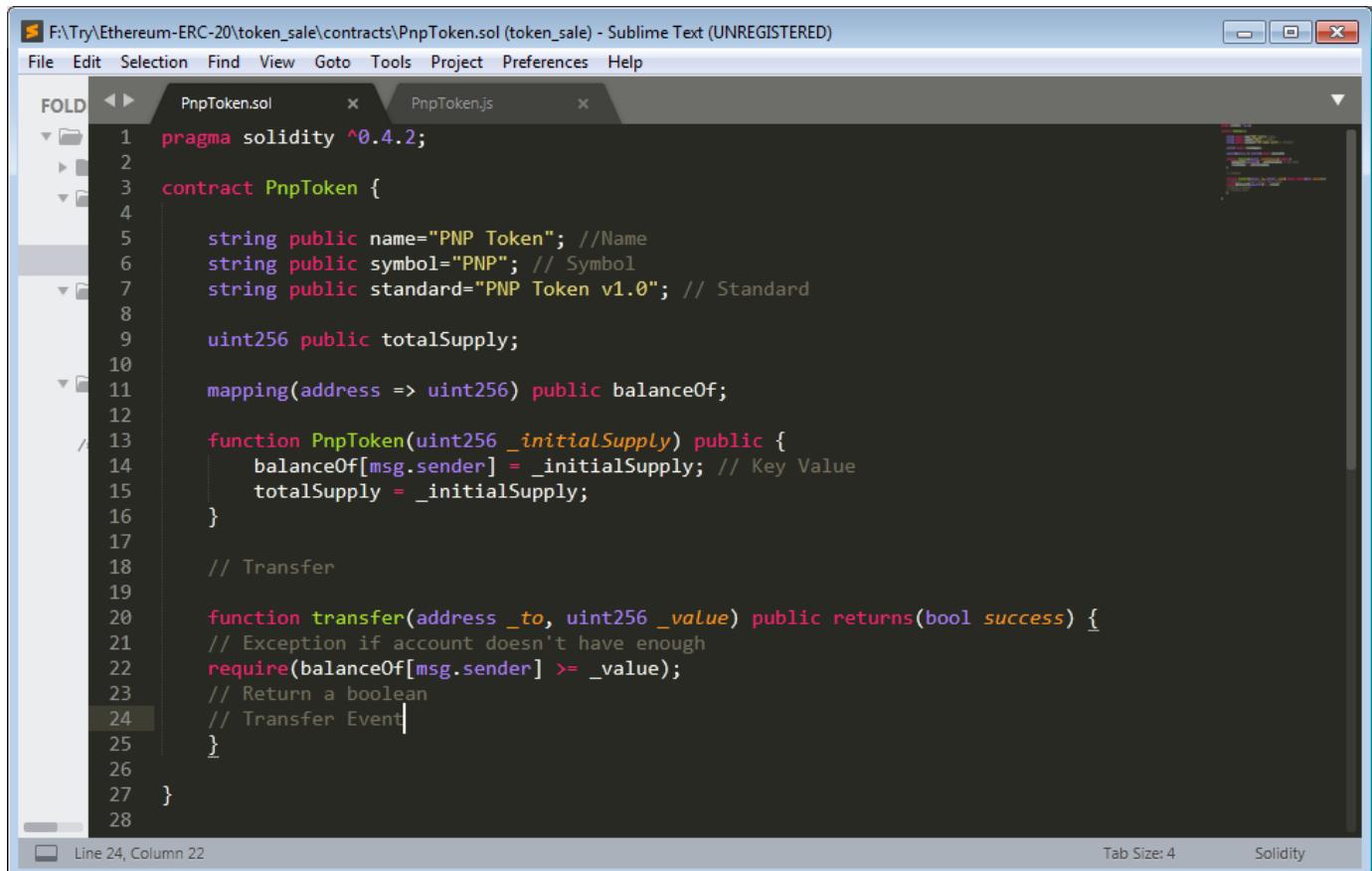
Contract: PnpToken
  1) initializes the contract with correct values (59ms)
  2) allocates the initial supply upon deployment
  3) transfers token ownership
    > No events were emitted

  2 passing (166ms)
  1 failing

1) Contract: PnpToken
    transfers token ownership:
      Assertion error message must contain revert
        at test\PnpToken.js:39:3
        at <anonymous>
        at process._tickCallback (internal/process/next_tick.js:189:7)
```

F:\Try\Ethereum-ERC-20\token_sale>

2. Read the balance from sender

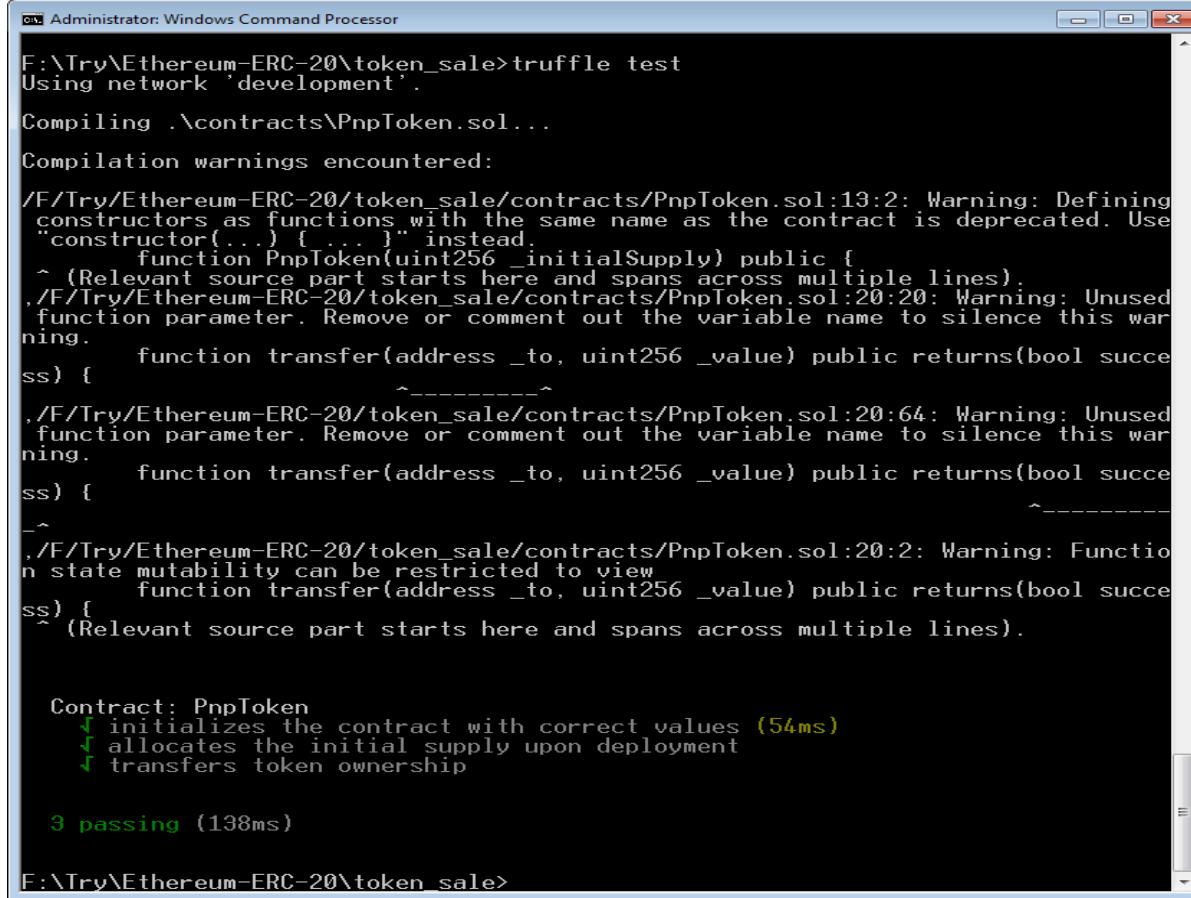


```

F:\Try\Ethereum-ERC-20\token_sale\contracts\PnpToken.sol (token_sale) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLD PnpToken.sol x PnpToken.js x
1 pragma solidity ^0.4.2;
2
3 contract PnpToken {
4
5     string public name="PNP Token"; //Name
6     string public symbol="PNP"; // Symbol
7     string public standard="PNP Token v1.0"; // Standard
8
9     uint256 public totalSupply;
10
11    mapping(address => uint256) public balanceOf;
12
13    function PnpToken(uint256 _initialSupply) public {
14        balanceOf[msg.sender] = _initialSupply; // Key Value
15        totalSupply = _initialSupply;
16    }
17
18    // Transfer
19
20    function transfer(address _to, uint256 _value) public returns(bool success) {
21        // Exception if account doesn't have enough
22        require(balanceOf[msg.sender] >= _value);
23        // Return a boolean
24        // Transfer Event
25    }
26
27 }
28

```

Line 24, Column 22 Tab Size: 4 Solidity



```

Administrator: Windows Command Processor
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:

/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:13:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){...}" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:20:20: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war
ning.
    function transfer(address _to, uint256 _value) public returns(bool succe
ss) {
^_____
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:20:64: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war
ning.
    function transfer(address _to, uint256 _value) public returns(bool succe
ss) {
^_____
^
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:20:2: Warning: Func
tion state mutability can be restricted to view
    function transfer(address _to, uint256 _value) public returns(bool succe
ss) {
^ (Relevant source part starts here and spans across multiple lines).

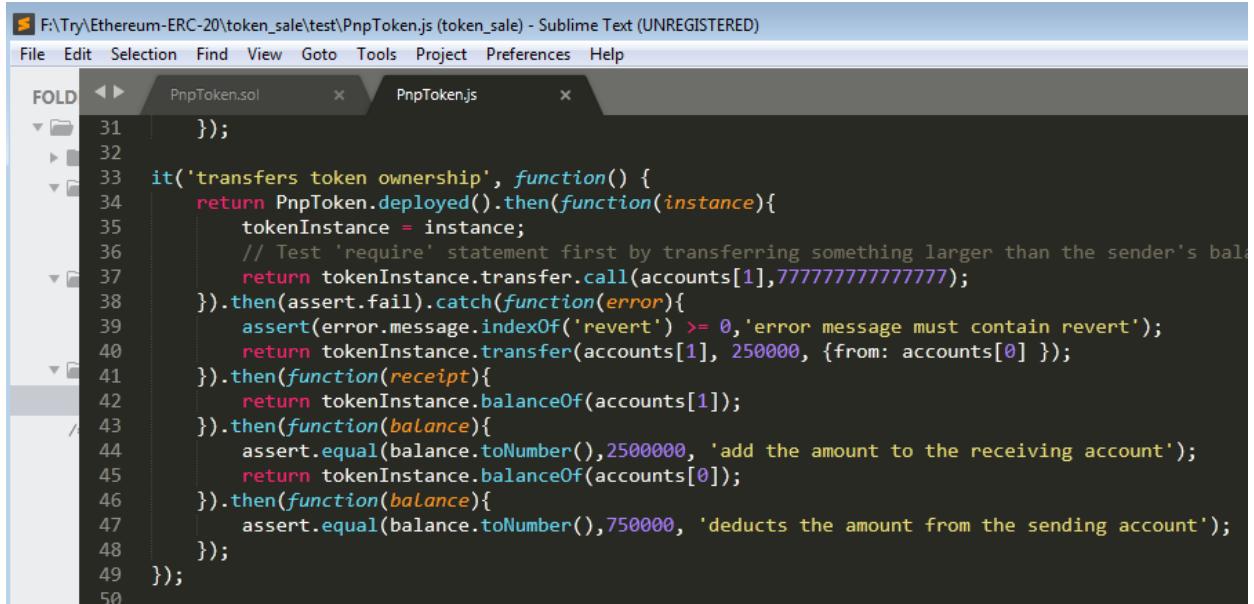
Contract: PnpToken
  initializes the contract with correct values (54ms)
  allocates the initial supply upon deployment
  transfers token ownership

3 passing (138ms)

F:\Try\Ethereum-ERC-20\token_sale>

```

3. Test the code



```

FATry\Ethereum-ERC-20\token_sale\test\PnPToken.js (token_sale) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLD PnPToken.sol PnPToken.js
31     });
32
33 it('transfers token ownership', function() {
34     return PnPToken.deployed().then(function(instance){
35         tokenInstance = instance;
36         // Test 'require' statement first by transferring something larger than the sender's balance
37         return tokenInstance.transfer.call(accounts[1], 777777777777777);
38     }).then(assert.fail).catch(function(error){
39         assert(error.message.indexOf('revert') >= 0, 'error message must contain revert');
40         return tokenInstance.transfer(accounts[1], 250000, {from: accounts[0]} );
41     }).then(function(receipt){
42         return tokenInstance.balanceOf(accounts[1]);
43     }).then(function(balance){
44         assert.equal(balance.toNumber(), 2500000, 'add the amount to the receiving account');
45         return tokenInstance.balanceOf(accounts[0]);
46     }).then(function(balance){
47         assert.equal(balance.toNumber(), 750000, 'deducts the amount from the sending account');
48     });
49 });
50

```

```
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.
```

```
Compiling .\contracts\PnPToken.sol...
```

```

Contract: PnPToken
  ✓ initializes the contract with correct values (85ms)
  ✓ allocates the initial supply upon deployment
1) transfers token ownership
  > No events were emitted

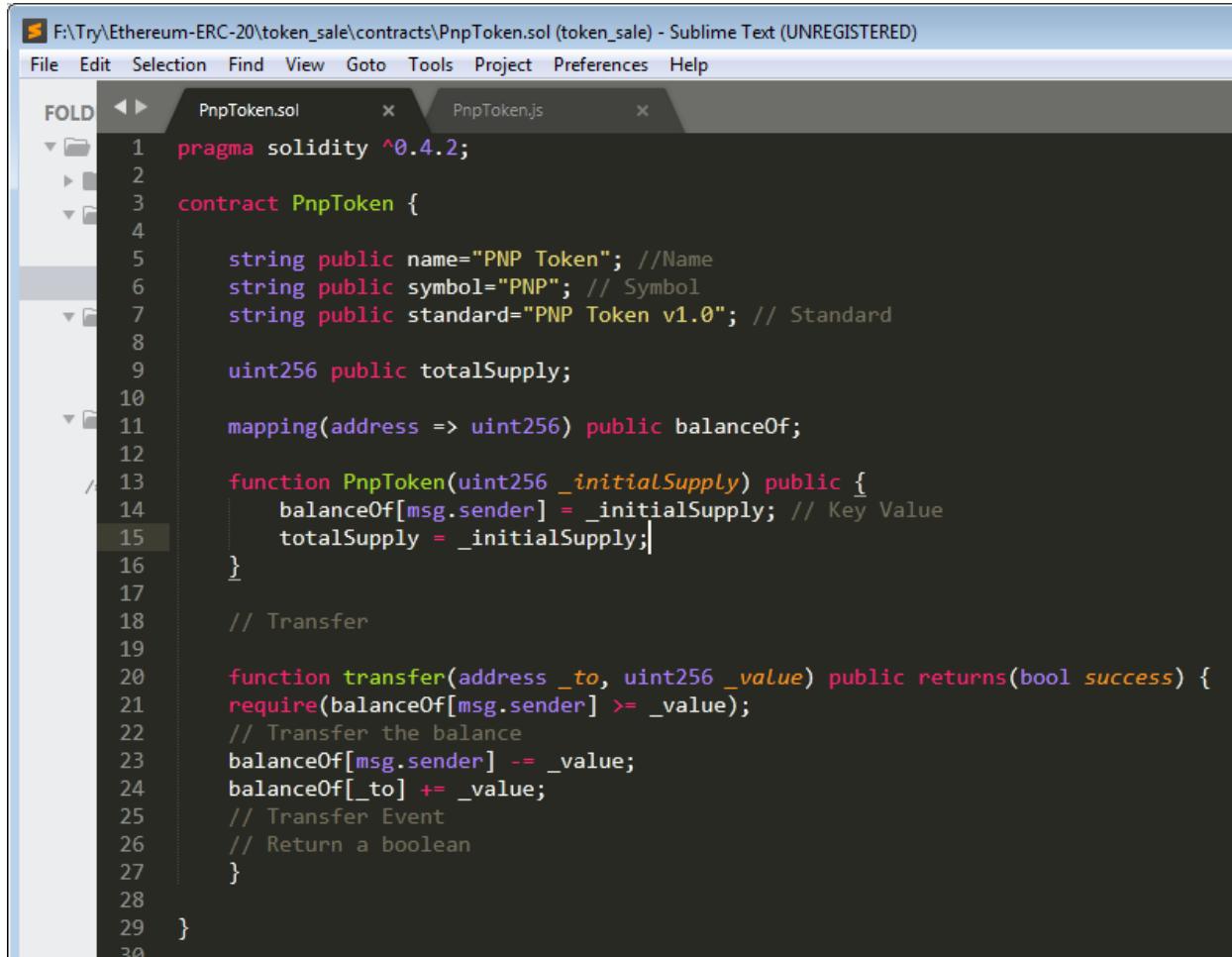
2 passing (260ms)
1 failing

1) Contract: PnPToken
   transfers token ownership:
     add the amount to the receiving account
     + expected - actual
       -0
       +2500000

     at test\PnPToken.js:44:10
     at <anonymous>
     at process._tickCallback (internal/process/next_tick.js:189:7)

```

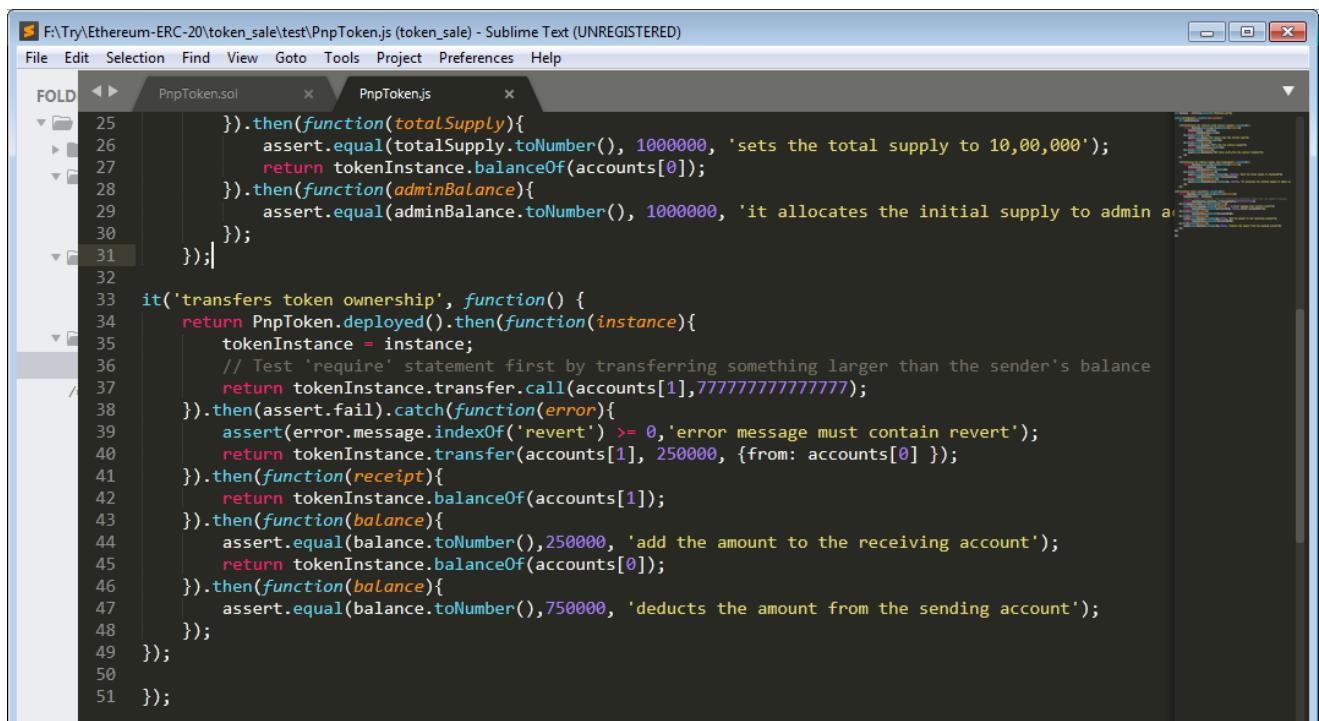
4. Modify the code as below (Transfer the balance)



```

1 pragma solidity ^0.4.2;
2
3 contract PnpToken {
4
5     string public name="PNP Token"; //Name
6     string public symbol="PNP"; // Symbol
7     string public standard="PNP Token v1.0"; // Standard
8
9     uint256 public totalSupply;
10
11    mapping(address => uint256) public balanceOf;
12
13    function PnpToken(uint256 _initialSupply) public {
14        balanceOf[msg.sender] = _initialSupply; // Key Value
15        totalSupply = _initialSupply;
16    }
17
18    // Transfer
19
20    function transfer(address _to, uint256 _value) public returns(bool success) {
21        require(balanceOf[msg.sender] >= _value);
22        // Transfer the balance
23        balanceOf[msg.sender] -= _value;
24        balanceOf[_to] += _value;
25        // Transfer Event
26        // Return a boolean
27    }
28
29 }
30

```



```

25 }).then(function(totalSupply){
26     assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 10,00,000');
27     return tokenInstance.balanceOf(accounts[0]);
28 }).then(function(adminBalance){
29     assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the initial supply to admin account');
30 });
31 });
32
33 it('transfers token ownership', function() {
34     return PnpToken.deployed().then(function(instance){
35         tokenInstance = instance;
36         // Test 'require' statement first by transferring something larger than the sender's balance
37         return tokenInstance.transfer.call(accounts[1],7777777777777777);
38     }).then(assert.fail).catch(function(error){
39         assert(error.message.indexOf('revert') >= 0,'error message must contain revert');
40         return tokenInstance.transfer(accounts[1], 250000, {from: accounts[0]} );
41     }).then(function(receipt){
42         return tokenInstance.balanceOf(accounts[1]);
43     }).then(function(balance){
44         assert.equal(balance.toNumber(),250000, 'add the amount to the receiving account');
45         return tokenInstance.balanceOf(accounts[0]);
46     }).then(function(balance){
47         assert.equal(balance.toNumber(),750000, 'deducts the amount from the sending account');
48     });
49 });
50
51 });

```

```
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:

/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:13:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){...}" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:20:64: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war
ning.
        function transfer(address _to, uint256 _value) public returns(bool succe
ss) {
^_____
-^

Contract: PnpToken
  ↓ initializes the contract with correct values (61ms)
  ↓ allocates the initial supply upon deployment
  ↓ transfers token ownership (99ms)

3 passing (227ms)

F:\Try\Ethereum-ERC-20\token_sale>
```

5. Modify the code as below (Transfer Event)

```
P:\Ethereum-ERC-20\token_sale\test\PnpToken.js (token_sale) - Sublime Text (UNREGISTERED)
dit Selection Find View Goto Tools Project Preferences Help
PnPToken.sol x PnPToken.js x
28     }).then(function(adminBalance){
29         assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the initial supply to admin account');
30     });
31 });
32
33 it('transfers token ownership', function() {
34     return PnpToken.deployed().then(function(instance){
35         tokenInstance = instance;
36         // Test 'require' statement first by transferring something larger than the sender's balance
37         return tokenInstance.transfer.call(accounts[1], 777777777777777);
38     }).then(assert.fail).catch(function(error){
39         assert(error.message.indexOf('revert') >= 0, 'error message must contain revert');
40         return tokenInstance.transfer(accounts[1], 250000, {from: accounts[0]});
41     }).then(function(receipt){
42         assert.equal(receipt.logs.length,1,'triggers one event');
43         assert.equal(receipt.logs[0].event,'Transfer','should be the "Transfer" event');
44         assert.equal(receipt.logs[0].args._from, accounts[0], 'logs the account the tokens are transferred from');
45         assert.equal(receipt.logs[0].args._to, accounts[1], 'logs the account the tokens are transferred to');
46         assert.equal(receipt.logs[0].args._value, 250000, 'logs the transfer amount');
47         return tokenInstance.balanceOf(accounts[1]);
48     }).then(function(balance){
49         assert.equal(balance.toNumber(),250000, 'add the amount to the receiving account');
50         return tokenInstance.balanceOf(accounts[0]);
51     }).then(function(balance){
52         assert.equal(balance.toNumber(),750000, 'deducts the amount from the sending account');
53     });
54 });
55 });
56 });

57});
```

y\Ethereum-ERC-20\token_sale\contracts\PnpToken.sol (token_sale) - Sublime Text (UNREGISTERED)

dit Selection Find View Goto Tools Project Preferences Help

```

1 pragma solidity ^0.4.2;
2
3 contract PnpToken {
4
5     string public name="PNP Token"; //Name
6     string public symbol="PNP"; // Symbol
7     string public standard="PNP Token v1.0"; // Standard
8     uint256 public totalSupply;
9
10    event Transfer(
11        address indexed _from,
12        address indexed _to,
13        uint256 _value
14    );
15
16
17    mapping(address => uint256) public balanceOf;
18
19    function PnpToken(uint256 _initialSupply) public {
20        balanceOf[msg.sender] = _initialSupply; // Key Value
21        totalSupply = _initialSupply;
22    }
23
24    // Transfer
25
26    function transfer(address _to, uint256 _value) public returns(bool success) {
27        require(balanceOf[msg.sender] >= _value);
28        // Transfer the balance
29        balanceOf[msg.sender] -= _value;
30        balanceOf[_to] += _value;
31        // Transfer Event
32        Transfer(msg.sender, _to, _value);
33        // Return a boolean
34    }
35
36}
37

```

Administrator: Windows Command Processor

```

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:

/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:19:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){...}" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:32:2: Warning: Invoking
events without "emit" prefix is deprecated.
    Transfer(msg.sender, _to, _value);

./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:26:64: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war-
ning.
    function transfer(address _to, uint256 _value) public returns(bool succe
ss) {
^-----^

Contract: PnpToken
  initializes the contract with correct values (54ms)
  allocates the initial supply upon deployment
  transfers token ownership (110ms)

3 passing (241ms)

F:\Try\Ethereum-ERC-20\token_sale>

```

6. Modify the code as below (take care of syntax)

The screenshot shows a code editor window with two tabs at the top: "PnpToken.sol" and "PnpToken.js". The "PnpToken.sol" tab is active, displaying the following Solidity code:

```
t Selection Find View Goto Tools Project Preferences Help
◀ ▶ PnpToken.sol • PnpToken.js •
1 pragma solidity ^0.4.2;
2
3 contract PnpToken {
4
5     string public name="PNP Token"; //Name
6     string public symbol="PNP"; // Symbol
7     string public standard="PNP Token v1.0"; // Standard
8     uint256 public totalSupply;
9
10    event Transfer(
11        address indexed _from,
12        address indexed _to,
13        uint256 _value
14    );
15
16
17    mapping(address => uint256) public balanceOf;
18
19    function PnpToken(uint256 _initialSupply) public {
20        balanceOf[msg.sender] = _initialSupply; // Key Value
21        totalSupply = _initialSupply;
22    }
23
24    // Transfer
25
26    function transfer(address _to, uint256 _value) public returns(bool success) {
27        require(balanceOf[msg.sender] >= _value);
28        // Transfer the balance
29        balanceOf[msg.sender] -= _value;
30        balanceOf[_to] += _value;
31        // Transfer Event
32        Transfer(msg.sender, _to, _value);
33        // Return a boolean
34        return true;
35    }
36}
37
38
39
40
```

```

 1 var PnpToken = artifacts.require("./PnpToken.sol");
 2
 3 contract('PnpToken', function (accounts) {
 4     var tokenInstance;
 5
 6     it('initializes the contract with correct values', function() {
 7         return PnpToken.deployed().then(function(instance){
 8             tokenInstance = instance;
 9             return tokenInstance.name();
10         }).then(function(name){
11             assert.equal(name,'PNP Token','has the correct name');
12             return tokenInstance.symbol();
13         }).then(function(symbol){
14             assert.equal(symbol,'PNP','has the correct symbol');
15             return tokenInstance.standard();
16         }).then(function(standard){
17             assert.equal(standard,'PNP Token v1.0','has the correct standard');
18         });
19     });
20
21     it('allocates the initial supply upon deployment', function() {
22         return PnpToken.deployed().then(function(instance){
23             tokenInstance = instance;
24             return tokenInstance.totalSupply();
25         }).then(function(totalSupply){
26             assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 10,00,000');
27             return tokenInstance.balanceOf(accounts[0]);
28         }).then(function(adminBalance){
29             assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the initial supply to admin account');
30         });
31     });
32
33     it('transfers token ownership', function() {
34         return PnpToken.deployed().then(function(instance){
35             tokenInstance = instance;
36             // Test 'require' statement first by transferring something larger than the sender's balance
37             return tokenInstance.transfer.call(accounts[1],7777777777777777);
38         }).then(assert.fail).catch(function(error){
39             assert(error.message.indexOf('revert') >= 0,'error message must contain revert');
40             return tokenInstance.transfer.call(accounts[1], 250000, { from: accounts[0] });
41         }).then(function(success){
42             assert.equal(success,true,'it returns true');
43             return tokenInstance.transfer(accounts[1], 250000, {from: accounts[0] });
44         }).then(function(receipt){
45             assert.equal(receipt.logs.length, 1,'triggers one event');
46             assert.equal(receipt.logs[0].event,'Transfer','should be the "Transfer" event');
47             assert.equal(receipt.logs[0].args._from, accounts[0], 'logs the account the tokens are transferred from');
48             assert.equal(receipt.logs[0].args._to, accounts[1], 'logs the account the tokens are transferred to');
49             assert.equal(receipt.logs[0].args._value, 250000, 'logs the transfer amount');
50             return tokenInstance.balanceOf(accounts[1]);
51         }).then(function(balance){
52             assert.equal(balance.toNumber(),250000, 'add the amount to the receiving account');
53             return tokenInstance.balanceOf(accounts[0]);
54         }).then(function(balance){
55             assert.equal(balance.toNumber(),750000, 'deducts the amount from the sending account');
56         });
57     });
58 });

```

```
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...

Compilation warnings encountered:

/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:19:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){ ... }" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:32:2: Warning: Invoking
events without "emit" prefix is deprecated.
    Transfer(msg.sender, _to, _value);
_____
_____|

Contract: PnpToken
  ✓ initializes the contract with correct values (52ms)
  ✓ allocates the initial supply upon deployment
  ✓ transfers token ownership (103ms)

3 passing (211ms)
```

Session 5: Approving Token Transfers

1. Add the code for Approve function

```
1 pragma solidity ^0.4.2;
2
3 contract PnpToken {
4
5     string public name="PNP Token"; //Name
6     string public symbol="PNP"; // Symbol
7     string public standard="PNP Token v1.0"; // Standard
8     uint256 public totalSupply;
9
10    event Transfer(
11         address indexed _from,
12         address indexed _to,
13         uint256 _value
14     );
15
16    // Transfer Event
17
18    mapping(address => uint256) public balanceOf;
19
20    // Allowance
21
22    function PnpToken(uint256 _initialSupply) public {
23        balanceOf[msg.sender] = _initialSupply; // Key Value
24        totalSupply = _initialSupply;
25    }
26
27    // Transfer
28    function transfer(address _to, uint256 _value) public returns(bool success) {
29        require(balanceOf[msg.sender] >= _value);
30        // Transfer the balance
31        balanceOf[msg.sender] -= _value;
32        balanceOf[_to] += _value;
33        // Transfer Event
34        Transfer(msg.sender, _to, _value);
35        // Return a boolean
36        return true;
37    }
38
39    // Approve
40    function approve (address _spender, uint256 _value) public returns(bool success)
41    {
42        return true;
43    }
44
45    // TransferFrom
46 }
```

```

PnpToken.sol      PnpToken.js
1 var PnpToken = artifacts.require("./PnpToken.sol");
2
3 contract('PnpToken', function (accounts) {
4     var tokenInstance;
5
6     it('initializes the contract with correct values', function() {
7         return PnpToken.deployed().then(function(instance){
8             tokenInstance = instance;
9             return tokenInstance.name();
10        }).then(function(name){
11            assert.equal(name,'PNP Token','has the correct name');
12            return tokenInstance.symbol();
13        }).then(function(symbol){
14            assert.equal(symbol,'PNP','has the correct symbol');
15            return tokenInstance.standard();
16        }).then(function(standard){
17            assert.equal(standard,'PNP Token v1.0','has the correct standard');
18        });
19    })
20
21    it('allocates the initial supply upon deployment', function() {
22        return PnpToken.deployed().then(function(instance){
23            tokenInstance = instance;
24            return tokenInstance.totalSupply();
25        }).then(function(totalSupply){
26            assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 10,00,000');
27            return tokenInstance.balanceOf(accounts[0]);
28        }).then(function(adminBalance){
29            assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the initial supply to admin account');
30        });
31    });
32
33    it('transfers token ownership', function() {
34        return PnpToken.deployed().then(function(instance){
35            tokenInstance = instance;
36            // Test 'require' statement first by transferring something larger than the sender's balance
37            return tokenInstance.transfer.call(accounts[1],777777777777777);
38        }).then(assert.fail).catch(function(error){
39            assert(error.message.indexOf('revert') >= 0,'error message must contain revert');
40            return tokenInstance.transfer.call(accounts[1], 250000, { from: accounts[0] });
41        }).then(function(success){
42            assert.equal(success,true,'it returns true');
43            return tokenInstance.transfer(accounts[1], 250000, {from: accounts[0] });
44        }).then(function(receipt){
45            assert.equal(receipt.logs.length, 1,'triggers one event');
46            assert.equal(receipt.logs[0].event,'Transfer','should be the "Transfer" event');
47            assert.equal(receipt.logs[0].args._from, accounts[0], 'logs the account the tokens are transferred from');
48            assert.equal(receipt.logs[0].args._to, accounts[1], 'logs the account the tokens are transferred to');
49            assert.equal(receipt.logs[0].args._value, 250000, 'logs the transfer amount');
50            return tokenInstance.balanceOf(accounts[1]);
51        }).then(function(balance){
52            assert.equal(balance.toNumber(),250000, 'add the amount to the receiving account');
53            return tokenInstance.balanceOf(accounts[0]);
54        }).then(function(balance){
55            assert.equal(balance.toNumber(),750000, 'deducts the amount from the sending account');
56        });
57    });
58
59    it('approves tokens for delegated transfer', function(){
60        return PnpToken.deployed().then(function(instance){
61            tokenInstance = instance;
62            return tokenInstance.approve.call(accounts[1], 100);
63        }).then(function(success){
64            assert.equal(success,true,'it returns true');
65        });
66    });
67
68 });

```

```
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:

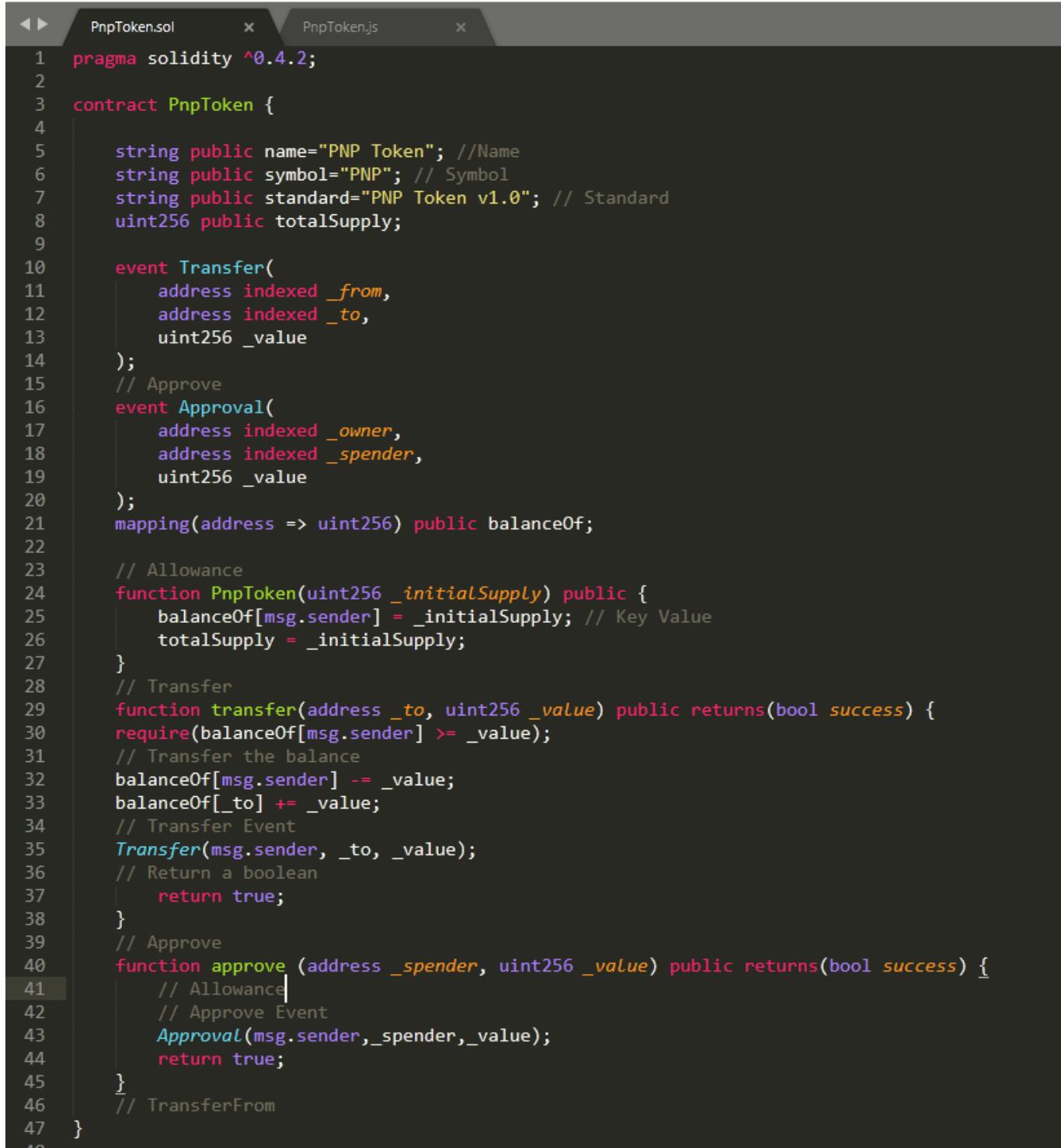
/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:22:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){ ... }" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
,/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:34:2: Warning: Invokin
g events without "emit" prefix is deprecated.
    Transfer(msg.sender, _to, _value);
^-----^

,/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:40:20: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war
ning.
    function approve (address _spender, uint256 _value) public returns(bool
success) {
^-----^
,/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:40:38: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war
ning.
    function approve (address _spender, uint256 _value) public returns(bool
success) {
^-----^
,/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:40:2: Warning: Function
state mutability can be restricted to pure
    function approve (address _spender, uint256 _value) public returns(bool
success) {
^ (Relevant source part starts here and spans across multiple lines).

Contract: PnpToken
  ✓ initializes the contract with correct values (51ms)
  ✓ allocates the initial supply upon deployment
  ✓ transfers token ownership (129ms)
  ✓ approves tokens for delegated transfer

  4 passing (249ms)
```

2. Add the code for Approval Event



```
1 pragma solidity ^0.4.2;
2
3 contract PnpToken {
4
5     string public name="PNP Token"; //Name
6     string public symbol="PNP"; // Symbol
7     string public standard="PNP Token v1.0"; // Standard
8     uint256 public totalSupply;
9
10    event Transfer(
11        address indexed _from,
12        address indexed _to,
13        uint256 _value
14    );
15    // Approve
16    event Approval(
17        address indexed _owner,
18        address indexed _spender,
19        uint256 _value
20    );
21    mapping(address => uint256) public balanceOf;
22
23    // Allowance
24    function PnpToken(uint256 _initialSupply) public {
25        balanceOf[msg.sender] = _initialSupply; // Key Value
26        totalSupply = _initialSupply;
27    }
28    // Transfer
29    function transfer(address _to, uint256 _value) public returns(bool success) {
30        require(balanceOf[msg.sender] >= _value);
31        // Transfer the balance
32        balanceOf[msg.sender] -= _value;
33        balanceOf[_to] += _value;
34        // Transfer Event
35        Transfer(msg.sender, _to, _value);
36        // Return a boolean
37        return true;
38    }
39    // Approve
40    function approve (address _spender, uint256 _value) public returns(bool success) {
41        // Allowance
42        // Approve Event
43        Approval(msg.sender,_spender,_value);
44        return true;
45    }
46    // TransferFrom
47 }
```

```

PnpToken.sol      PnpToken.js
1 var PnpToken = artifacts.require("./PnpToken.sol");
2
3 contract('PnpToken', function (accounts) {
4     var tokenInstance;
5
6     it('initializes the contract with correct values', function() {
7         return PnpToken.deployed().then(function(instance){
8             tokenInstance = instance;
9             return tokenInstance.name();
10        }).then(function(name){
11            assert.equal(name,'PNP Token','has the correct name');
12            return tokenInstance.symbol();
13        }).then(function(symbol){
14            assert.equal(symbol,'PNP','has the correct symbol');
15            return tokenInstance.standard();
16        }).then(function(stdard){
17            assert.equal(stdard,'PNP Token v1.0','has the correct standard');
18        });
19    });
20
21    it('allocates the initial supply upon deployment', function() {
22        return PnpToken.deployed().then(function(instance){
23            tokenInstance = instance;
24            return tokenInstance.totalSupply();
25        }).then(function(totalSupply){
26            assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 10,00,000');
27            return tokenInstance.balanceOf(accounts[0]);
28        }).then(function(adminBalance){
29            assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the initial supply to admin account');
30        });
31    });
32
33    it('transfers token ownership', function() {
34        return PnpToken.deployed().then(function(instance){
35            tokenInstance = instance;
36            // Test 'require' statement first by transferring something larger than the sender's balance
37            return tokenInstance.transfer.call(accounts[1],777777777777777);
38        }).then(assert.fail).catch(function(error){
39            assert(error.message.indexOf('revert') >= 0,'error message must contain revert');
40            return tokenInstance.transfer.call(accounts[1], 250000, { from: accounts[0] });
41        }).then(function(success){
42            assert.equal(success,true,'it returns true');
43            return tokenInstance.transfer(accounts[1], 250000, {from: accounts[0] });
44        }).then(function(receipt){
45            assert.equal(receipt.logs.length, 1,'triggers one event');
46            assert.equal(receipt.logs[0].event,'Transfer','should be the "Transfer" event');
47            assert.equal(receipt.logs[0].args._from, accounts[0], 'logs the account the tokens are transferred from');
48            assert.equal(receipt.logs[0].args._to, accounts[1], 'logs the account the tokens are transferred to');
49            assert.equal(receipt.logs[0].args._value, 250000, 'logs the transfer amount');
50            return tokenInstance.balanceOf(accounts[1]);
51        }).then(function(balance){
52            assert.equal(balance.toNumber(),250000, 'add the amount to the receiving account');
53            return tokenInstance.balanceOf(accounts[0]);
54        }).then(function(balance){
55            assert.equal(balance.toNumber(),750000, 'deducts the amount from the sending account');
56        });
57    });
58
59    it('approves tokens for delegated transfer', function(){
60        return PnpToken.deployed().then(function(instance){
61            tokenInstance = instance;
62            return tokenInstance.approve.call(accounts[1], 100);
63        }).then(function(success){
64            assert.equal(success,true,'it returns true');
65            return tokenInstance.approve(accounts[1], 100); // It will create Transaction
66        }).then(function(receipt) {
67            assert.equal(receipt.logs.length, 1,'triggers one event');
68            assert.equal(receipt.logs[0].event,'Approval','should be the "Approval" event');
69            assert.equal(receipt.logs[0].args._owner, accounts[0], 'logs the account the tokens are authorized by');
70            assert.equal(receipt.logs[0].args._spender, accounts[1], 'logs the account the tokens are authorized to');
71            assert.equal(receipt.logs[0].args._value, 100, 'logs the transfer amount');
72        });
73    });
74
75 });

```

```
Administrator: Windows Command Processor

F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:

/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:28:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){...}" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
,/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:40:2: Warning: Invoking
events without "emit" prefix is deprecated.
    Transfer(msg.sender, _to, _value);
^-----_
,/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:50:3: Warning: Invoking
events without "emit" prefix is deprecated.
    Approval(msg.sender,_spender,_value);
^-----_

Contract: PnpToken
  ✓ initializes the contract with correct values (51ms)
  ✓ allocates the initial supply upon deployment
  ✓ transfers token ownership (117ms)
  ✓ approves tokens for delegated transfer (59ms)

  4 passing (297ms)

F:\Try\Ethereum-ERC-20\token_sale>
```

3. Add the code for Allowance

```
1 pragma solidity ^0.4.2;
2
3 contract PnpToken {
4     string public name="PNP Token"; //Name
5     string public symbol="PNP"; // Symbol
6     string public standard="PNP Token v1.0"; // Standard
7     uint256 public totalSupply;
8     event Transfer(
9         address indexed _from,
10        address indexed _to,
11        uint256 _value
12    );
13    // Approve
14    event Approval(
15        address indexed _owner,
16        address indexed _spender,
17        uint256 _value
18    );
19    mapping(address => uint256) public balanceOf;
20    mapping(address => mapping(address => uint256)) public allowance;
21    // Allowance
22    function PnpToken(uint256 _initialSupply) public {
23        balanceOf[msg.sender] = _initialSupply; // Key Value
24        totalSupply = _initialSupply;
25    }
26    // Transfer
27    function transfer(address _to, uint256 _value) public returns(bool success) {
28        require(balanceOf[msg.sender] >= _value);
29        balanceOf[msg.sender] -= _value; // Transfer the balance
30        balanceOf[_to] += _value;
31        Transfer(msg.sender, _to, _value); // Transfer Event
32        return true; // Return a boolean
33    }
34    // Approve
35    function approve (address _spender, uint256 _value) public returns(bool success) {
36        // Allowance
37        allowance[msg.sender][_spender] = _value;
38        Approval(msg.sender,_spender,_value); // Approve Event
39        return true;
40    }
41    // TransferFrom
42 }
```

```
◀ ▶ PnpToken.sol × PnpToken.js ×
1 var PnpToken = artifacts.require("./PnpToken.sol");
2
3 contract('PnpToken', function (accounts) {
4     var tokenInstance;
5
6     it('initializes the contract with correct values', function() {
7         return PnpToken.deployed().then(function(instance){
8             tokenInstance = instance;
9             return tokenInstance.name();
10        }).then(function(name){
11            assert.equal(name,'PNP Token','has the correct name');
12            return tokenInstance.symbol();
13        }).then(function(symbol){
14            assert.equal(symbol,'PNP','has the correct symbol');
15            return tokenInstance.standard();
16        }).then(function(standard){
17            assert.equal(standard,'PNP Token v1.0','has the correct standard');
18        });
19    })
20
21    it('allocates the initial supply upon deployment', function() {
22        return PnpToken.deployed().then(function(instance){
23            tokenInstance = instance;
24            return tokenInstance.totalSupply();
25        }).then(function(totalSupply){
26            assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 10,00,000');
27            return tokenInstance.balanceOf(accounts[0]);
28        }).then(function(adminBalance){
29            assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the initial supply to admin account');
30        });
31    });
32});
```

```
 31     });
 32
 33 ▼ it('transfers token ownership', function() {
 34   return PnpToken.deployed().then(function(instance){
 35     tokenInstance = instance;
 36     // Test 'require' statement first by transferring something larger than the sender's balance
 37     return tokenInstance.transfer.call(accounts[1],7777777777777777);
 38   }).then(assert.fail).catch(function(error){
 39     assert(error.message.indexOf('revert') >= 0,'error message must contain revert');
 40     return tokenInstance.transfer.call(accounts[1], 250000, { from: accounts[0] });
 41   }).then(function(success){
 42     assert.equal(success,true,'it returns true');
 43     return tokenInstance.transfer(accounts[1], 250000, {from: accounts[0] });
 44   }).then(function(receipt){
 45     assert.equal(receipt.logs.length, 1,'triggers one event');
 46     assert.equal(receipt.logs[0].event,'Transfer','should be the "Transfer" event');
 47     assert.equal(receipt.logs[0].args._from, accounts[0], 'logs the account the tokens are transferred from');
 48     assert.equal(receipt.logs[0].args._to, accounts[1], 'logs the account the tokens are transferred to');
 49     assert.equal(receipt.logs[0].args._value, 250000, 'logs the transfer amount');
 50     return tokenInstance.balanceOf(accounts[1]);
 51   }).then(function(balance){
 52     assert.equal(balance.toNumber(),250000, 'add the amount to the receiving account');
 53     return tokenInstance.balanceOf(accounts[0]);
 54   }).then(function(balance){
 55     assert.equal(balance.toNumber(),750000, 'deducts the amount from the sending account');
 56   });
 57 });
 58
 59 ▼ it('approves tokens for delegated transfer', function(){
 60   return PnpToken.deployed().then(function(instance){
 61     tokenInstance = instance;
 62     return tokenInstance.approve.call(accounts[1], 100);
 63   }).then(function(success){
 64     assert.equal(success,true,'it returns true');
 65     return tokenInstance.approve(accounts[1], 100, {from: accounts[0] }); // It will create Transaction
 66   }).then(function(receipt) {
 67     assert.equal(receipt.logs.length, 1,'triggers one event');
 68     assert.equal(receipt.logs[0].event,'Approval','should be the "Approval" event');
 69     assert.equal(receipt.logs[0].args._owner, accounts[0], 'logs the account the tokens are authorized by');
 70     assert.equal(receipt.logs[0].args._spender, accounts[1], 'logs the account the tokens are authorized to');
 71     assert.equal(receipt.logs[0].args._value, 100, 'logs the transfer amount');
 72     return tokenInstance.allowance(accounts[0], accounts[1]);
 73   }).then(function(allowance){
 74     assert.equal(allowance.toNumber(),100,'stores the allowance for delegated transfer');
 75   });
 76 });
 77 });
```

```
Administrator: Windows Command Processor

F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:

/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:22:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){...}" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
,/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:31:2: Warning: Invokin
g events without "emit" prefix is deprecated.
    Transfer(msg.sender, _to, _value); // Transfer Event
^-----^
,/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:38:3: Warning: Invokin
g events without "emit" prefix is deprecated.
    Approval(msg.sender,_spender,_value); // Approve Event
^-----^

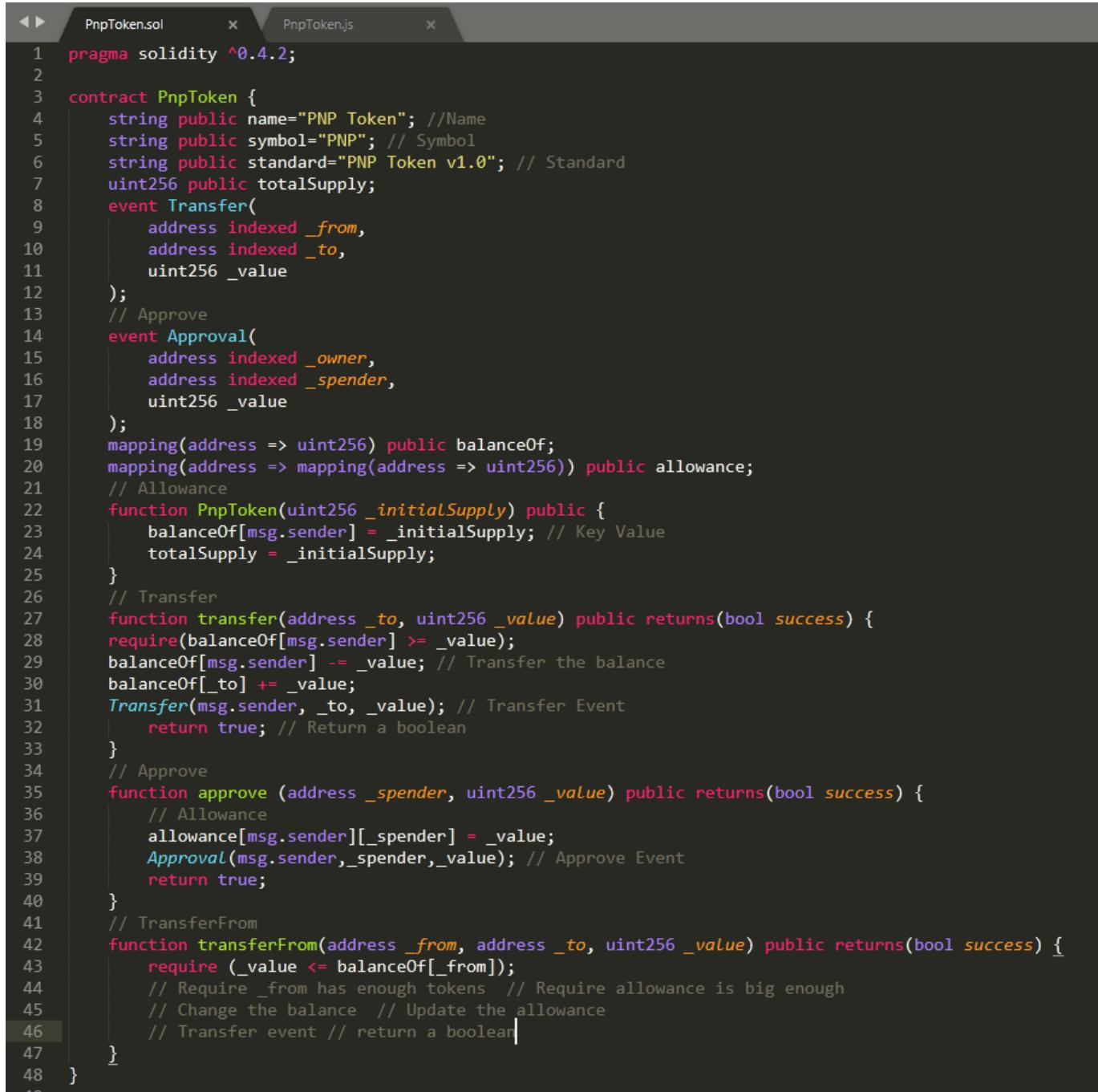
Contract: PnpToken
    ✓ initializes the contract with correct values (47ms)
    ✓ allocates the initial supply upon deployment
    ✓ transfers token ownership (124ms)
    ✓ approves tokens for delegated transfer (89ms)

4 passing (333ms)

F:\Try\Ethereum-ERC-20\token_sale>
```

Session 6: Delegated Token Transfers

1. Add TransferFrom function in the code



```
1 pragma solidity ^0.4.2;
2
3 contract PnPToken {
4     string public name="PNP Token"; //Name
5     string public symbol="PNP"; // Symbol
6     string public standard="PNP Token v1.0"; // Standard
7     uint256 public totalSupply;
8     event Transfer(
9         address indexed _from,
10        address indexed _to,
11        uint256 _value
12    );
13    // Approve
14    event Approval(
15        address indexed _owner,
16        address indexed _spender,
17        uint256 _value
18    );
19    mapping(address => uint256) public balanceOf;
20    mapping(address => mapping(address => uint256)) public allowance;
21    // Allowance
22    function PnPToken(uint256 _initialSupply) public {
23        balanceOf[msg.sender] = _initialSupply; // Key Value
24        totalSupply = _initialSupply;
25    }
26    // Transfer
27    function transfer(address _to, uint256 _value) public returns(bool success) {
28        require(balanceOf[msg.sender] >= _value);
29        balanceOf[msg.sender] -= _value; // Transfer the balance
30        balanceOf[_to] += _value;
31        Transfer(msg.sender, _to, _value); // Transfer Event
32        return true; // Return a boolean
33    }
34    // Approve
35    function approve (address _spender, uint256 _value) public returns(bool success) {
36        // Allowance
37        allowance[msg.sender][_spender] = _value;
38        Approval(msg.sender,_spender,_value); // Approve Event
39        return true;
40    }
41    // TransferFrom
42    function transferFrom(address _from, address _to, uint256 _value) public returns(bool success) {
43        require (_value <= balanceOf[_from]);
44        // Require _from has enough tokens // Require allowance is big enough
45        // Change the balance // Update the allowance
46        // Transfer event // return a boolean
47    }
48 }
```

```

1 var PnpToken = artifacts.require("./PnpToken.sol");
2
3 contract('PnpToken', function (accounts) {
4     var tokenInstance;
5
6     it('initializes the contract with correct values', function() {
7         return PnpToken.deployed().then(function(instance){
8             tokenInstance = instance;
9             return tokenInstance.name();
10        }).then(function(name){
11            assert.equal(name,'PNP Token','has the correct name');
12            return tokenInstance.symbol();
13        }).then(function(symbol){
14            assert.equal(symbol,'PNP','has the correct symbol');
15            return tokenInstance.standard();
16        }).then(function(standard){
17            assert.equal(standard,'PNP Token v1.0','has the correct standard');
18        });
19    })
20
21    it('allocates the initial supply upon deployment', function() {
22        return PnpToken.deployed().then(function(instance){
23            tokenInstance = instance;
24            return tokenInstance.totalSupply();
25        }).then(function(totalSupply){
26            assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 10,00,000');
27            return tokenInstance.balanceOf(accounts[0]);
28        }).then(function(adminBalance){
29            assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the initial supply to admin account');
30        });
31    });
32
33    it('transfers token ownership', function() {
34        return PnpToken.deployed().then(function(instance){
35            tokenInstance = instance;
36            // Test 'require' statement first by transferring something larger than the sender's balance
37            return tokenInstance.transfer.call(accounts[1],777777777777777);
38        }).then(assert.fail).catch(function(error){
39            assert(error.message.indexOf('revert') >= 0,'error message must contain revert');
40            return tokenInstance.transfer.call(accounts[1], 250000, { from: accounts[0] });
41        }).then(function(success){
42            assert.equal(success,true,'it returns true');
43            return tokenInstance.transfer(accounts[1], 250000, {from: accounts[0] });
44        }).then(function(receipt){
45            assert.equal(receipt.logs.length, 1,'triggers one event');
46            assert.equal(receipt.logs[0].event,'Transfer','should be the "Transfer" event');
47            assert.equal(receipt.logs[0].args._from, accounts[0], 'logs the account the tokens are transferred from');
48            assert.equal(receipt.logs[0].args._to, accounts[1], 'logs the account the tokens are transferred to');
49            assert.equal(receipt.logs[0].args._value, 250000, 'logs the transfer amount');
50            return tokenInstance.balanceOf(accounts[1]);
51        }).then(function(balance){
52            assert.equal(balance.toNumber(),250000, 'add the amount to the receiving account');
53            return tokenInstance.balanceOf(accounts[0]);
54        }).then(function(balance){
55            assert.equal(balance.toNumber(),750000, 'deducts the amount from the sending account');
56        });
57    });
58
59    it('approves tokens for delegated transfer', function(){
60        return PnpToken.deployed().then(function(instance){
61            tokenInstance = instance;
62            return tokenInstance.approve.call(accounts[1], 100);
63        }).then(function(success){
64            assert.equal(success,true,'it returns true');
65            return tokenInstance.approve(accounts[1], 100, {from: accounts[0]}); // It will create Transaction
66        }).then(function(receipt) {
67            assert.equal(receipt.logs.length, 1,'triggers one event');
68            assert.equal(receipt.logs[0].event,'Approval','should be the "Approval" event');
69            assert.equal(receipt.logs[0].args._owner, accounts[0], 'logs the account the tokens are authorized by');
70            assert.equal(receipt.logs[0].args._spender, accounts[1], 'logs the account the tokens are authorized to');
71            assert.equal(receipt.logs[0].args._value, 100, 'logs the transfer amount');
72            return tokenInstance.allowance(accounts[0], accounts[1]);
73        }).then(function(allowance){
74            assert.equal(allowance.toNumber(),100,'stores the allowance for delegated transfer');
75        });
76    });
}

```

```

77
78 it('handles delegated token transfer', function(){
79   return PnpToken.deployed().then(function(instance){
80     tokenInstance = instance;
81     fromAccount = accounts[2];
82     toAccount = accounts[3];
83     spendingAccount = accounts[4];
84     // Transfer some tokens to fromAccount
85     return tokenInstance.transfer(fromAccount, 100, { from: accounts[0] });
86   }).then(function(receipt){
87     // Approve spendingAccount to spend 10 tokens fromAccount
88     return tokenInstance.approve(spendingAccount, 10, { from: fromAccount });
89   }).then(function(receipt){
90     // Try transferring larger than the sender's balance
91     return tokenInstance.transferFrom(fromAccount, toAccount, 7777, { from: spendingAccount });
92   }).then(assert.fail).catch(function(error){
93     assert(error.message.indexOf('revert') >= 0, 'cannot tranfer value larger than balance');
94   });
95 });
96 });
97

```

The screenshot shows a Windows Command Processor window titled "Administrator: Windows Command Processor". The command entered is "F:\Try\Ethereum-ERC-20\token_sale>truffle test". The output displays the results of the Truffle test, which includes compilation warnings for the PnpToken.sol contract. The warnings are related to deprecated practices such as defining constructors as functions, using "emit" prefixes for events, and unused function parameters. The test results show 5 passing tests in 538ms.

```

Administrator: Windows Command Processor
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compilation warnings encountered:

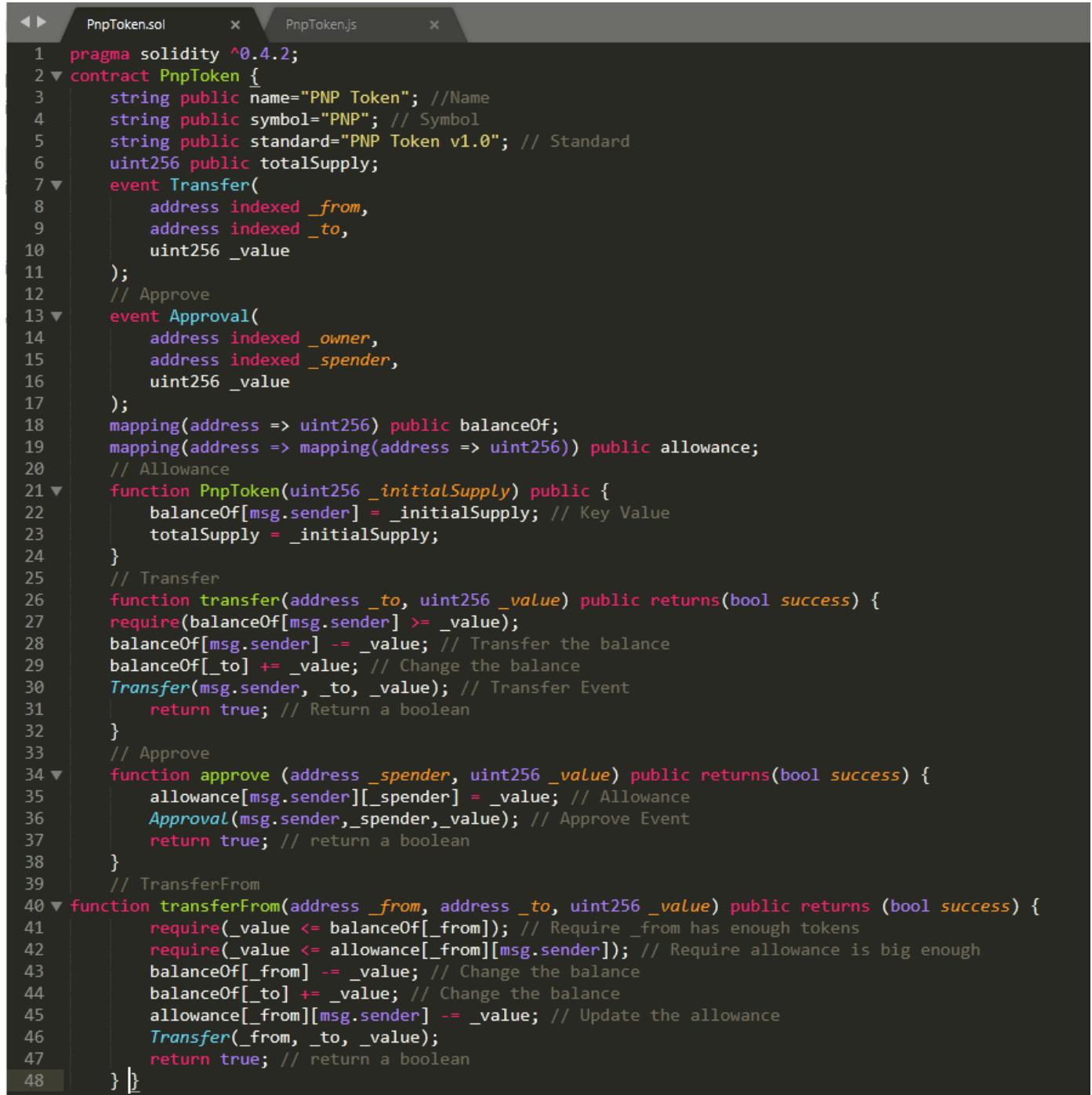
/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:22:2: Warning: Defining
constructors as functions with the same name as the contract is deprecated. Use
"constructor(...){ ... }" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:31:2: Warning: Invoking
events without "emit" prefix is deprecated.
    Transfer(msg.sender, _to, _value); // Transfer Event
^_____
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:38:3: Warning: Invoking
events without "emit" prefix is deprecated.
    Approval(msg.sender,_spender,_value); // Approve Event
^_____
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:42:39: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war-
ning.
    function transferFrom(address _from, address _to, uint256 _value) public
returns(bool success) {
^_____
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:42:83: Warning: Unused
function parameter. Remove or comment out the variable name to silence this war-
ning.
    function transferFrom(address _from, address _to, uint256 _value) public
returns(bool success) {
^_____
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:42:2: Warning: Function
state mutability can be restricted to view
    function transferFrom(address _from, address _to, uint256 _value) public
returns(bool success) {
^ (Relevant source part starts here and spans across multiple lines).

Contract: PnpToken
  ✓ initializes the contract with correct values (51ms)
  ✓ allocates the initial supply upon deployment
  ✓ transfers token ownership (161ms)
  ✓ approves tokens for delegated transfer (64ms)
  ✓ handles delegated token transfer (163ms)

5 passing (538ms)

```

2. Modify the code as below



The screenshot shows a code editor with two tabs: 'PnpToken.sol' and 'PnpToken.js'. The 'PnpToken.sol' tab is active, displaying the Solidity code for a ERC-20 token. The code defines a contract named PnPToken with various functions for transferring tokens, approving spenders, and checking balances. It also includes event definitions for Transfer and Approval. The code uses standard Solidity syntax with comments explaining the logic.

```
1 pragma solidity ^0.4.2;
2 contract PnPToken {
3     string public name="PnP Token"; //Name
4     string public symbol="PNP"; // Symbol
5     string public standard="PNP Token v1.0"; // Standard
6     uint256 public totalSupply;
7     event Transfer(
8         address indexed _from,
9         address indexed _to,
10        uint256 _value
11    );
12    // Approve
13    event Approval(
14        address indexed _owner,
15        address indexed _spender,
16        uint256 _value
17    );
18    mapping(address => uint256) public balanceOf;
19    mapping(address => mapping(address => uint256)) public allowance;
20    // Allowance
21    function PnPToken(uint256 _initialSupply) public {
22        balanceOf[msg.sender] = _initialSupply; // Key Value
23        totalSupply = _initialSupply;
24    }
25    // Transfer
26    function transfer(address _to, uint256 _value) public returns(bool success) {
27        require(balanceOf[msg.sender] >= _value);
28        balanceOf[msg.sender] -= _value; // Transfer the balance
29        balanceOf[_to] += _value; // Change the balance
30        Transfer(msg.sender, _to, _value); // Transfer Event
31        return true; // Return a boolean
32    }
33    // Approve
34    function approve (address _spender, uint256 _value) public returns(bool success) {
35        allowance[msg.sender][_spender] = _value; // Allowance
36        Approval(msg.sender,_spender,_value); // Approve Event
37        return true; // return a boolean
38    }
39    // TransferFrom
40    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
41        require(_value <= balanceOf[_from]); // Require _from has enough tokens
42        require(_value <= allowance[_from][msg.sender]); // Require allowance is big enough
43        balanceOf[_from] -= _value; // Change the balance
44        balanceOf[_to] += _value; // Change the balance
45        allowance[_from][msg.sender] -= _value; // Update the allowance
46        Transfer(_from, _to, _value);
47        return true; // return a boolean
48    }
}
```

```
1 var PnpToken = artifacts.require("./PnpToken.sol");
2
3 contract('PnpToken', function (accounts) {
4     var tokenInstance;
5
6     it('initializes the contract with correct values', function() {
7         return PnpToken.deployed().then(function(instance){
8             tokenInstance = instance;
9             return tokenInstance.name();
10        }).then(function(name){
11            assert.equal(name,'PNP Token','has the correct name');
12            return tokenInstance.symbol();
13        }).then(function(symbol){
14            assert.equal(symbol,'PNP','has the correct symbol');
15            return tokenInstance.standard();
16        }).then(function(standard){
17            assert.equal(standard,'PNP Token v1.0','has the correct standard');
18        });
19    })
20
21    it('allocates the initial supply upon deployment', function() {
22        return PnpToken.deployed().then(function(instance){
23            tokenInstance = instance;
24            return tokenInstance.totalSupply();
25        }).then(function(totalSupply){
26            assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 10,00,000');
27            return tokenInstance.balanceOf(accounts[0]);
28        }).then(function(adminBalance){
29            assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the initial supply to admin account');
30        });
31    });
32});
```

```

33 it('transfers token ownership', function() {
34     return PnpToken.deployed().then(function(instance){
35         tokenInstance = instance;
36         // Test 'require' statement first by transferring something larger than the sender's balance
37         return tokenInstance.transfer.call(accounts[1],7777777777777777);
38     }).then(assert.fail).catch(function(error){
39         assert(error.message.indexOf('revert') >= 0,'error message must contain revert');
40         return tokenInstance.transfer.call(accounts[1], 250000, { from: accounts[0] });
41     }).then(function(success){
42         assert.equal(success,true,'it returns true');
43         return tokenInstance.transfer(accounts[1], 250000, {from: accounts[0] });
44     }).then(function(receipt){
45         assert.equal(receipt.logs.length, 1,'triggers one event');
46         assert.equal(receipt.logs[0].event,'Transfer','should be the "Transfer" event');
47         assert.equal(receipt.logs[0].args._from, accounts[0], 'logs the account the tokens are transferred from');
48         assert.equal(receipt.logs[0].args._to, accounts[1], 'logs the account the tokens are transferred to');
49         assert.equal(receipt.logs[0].args._value, 250000, 'logs the transfer amount');
50         return tokenInstance.balanceOf(accounts[1]);
51     }).then(function(balance){
52         assert.equal(balance.toNumber(),250000, 'add the amount to the receiving account');
53         return tokenInstance.balanceOf(accounts[0]);
54     }).then(function(balance){
55         assert.equal(balance.toNumber(),750000, 'deducts the amount from the sending account');
56     });
57 });
58
59 it('approves tokens for delegated transfer', function(){
60     return PnpToken.deployed().then(function(instance){
61         tokenInstance = instance;
62         return tokenInstance.approve.call(accounts[1], 100);
63     }).then(function(success){
64         assert.equal(success,true,'it returns true');
65         return tokenInstance.approve(accounts[1], 100, {from: accounts[0] }); // It will create Transaction
66     }).then(function(receipt) {
67         assert.equal(receipt.logs.length, 1,'triggers one event');
68         assert.equal(receipt.logs[0].event,'Approval','should be the "Approval" event');
69         assert.equal(receipt.logs[0].args._owner, accounts[0], 'logs the account the tokens are authorized by');
70         assert.equal(receipt.logs[0].args._spender, accounts[1], 'logs the account the tokens are authorized to');
71         assert.equal(receipt.logs[0].args._value, 100, 'logs the transfer amount');
72         return tokenInstance.allowance(accounts[0], accounts[1]);
73     }).then(function(allowance){
74         assert.equal(allowance.toNumber(),100,'stores the allowance for delegated transfer');
75     });
76 });
77

```

```
78 ▼ it('handles delegated token transfers', function() {
79 ▼   return PnpToken.deployed().then(function(instance) {
80     tokenInstance = instance;
81     fromAccount = accounts[2];
82     toAccount = accounts[3];
83     spendingAccount = accounts[4];
84     // Transfer some tokens to fromAccount
85     return tokenInstance.transfer(fromAccount, 100, { from: accounts[0] });
86   }).then(function(receipt) {
87     // Approve spendingAccount to spend 10 tokens from fromAccount
88     return tokenInstance.approve(spendingAccount, 10, { from: fromAccount });
89   }).then(function(receipt) {
90     // Try transferring something larger than the sender's balance
91     return tokenInstance.transferFrom(fromAccount, toAccount, 7777, { from: spendingAccount });
92   }).then(assert.fail).catch(function(error) {
93     assert(error.message.indexOf('revert') >= 0, 'cannot transfer value larger than balance');
94     // Try transferring something larger than the approved amount
95     return tokenInstance.transferFrom(fromAccount, toAccount, 20, { from: spendingAccount });
96   }).then(assert.fail).catch(function(error) {
97     assert(error.message.indexOf('revert') >= 0, 'cannot transfer value larger than approved amount');
98     return tokenInstance.transferFrom.call(fromAccount, toAccount, 10, { from: spendingAccount });
99   }).then(function(success) {
100    assert.equal(success, true);
101    return tokenInstance.transferFrom(fromAccount, toAccount, 10, { from: spendingAccount });
102  }).then(function(receipt) {
103    assert.equal(receipt.logs.length, 1, 'triggers one event');
104    assert.equal(receipt.logs[0].event, 'Transfer', 'should be the "Transfer" event');
105    assert.equal(receipt.logs[0].args._from, fromAccount, 'logs the account the tokens are transferred from');
106    assert.equal(receipt.logs[0].args._to, toAccount, 'logs the account the tokens are transferred to');
107    assert.equal(receipt.logs[0].args._value, 10, 'logs the transfer amount');
108    return tokenInstance.balanceOf(fromAccount);
109  }).then(function(balance) {
110    assert.equal(balance.toNumber(), 90, 'deducts the amount from the sending account');
111    return tokenInstance.balanceOf(toAccount);
112  }).then(function(balance) {
113    assert.equal(balance.toNumber(), 10, 'adds the amount from the receiving account');
114    return tokenInstance.allowance(fromAccount, spendingAccount);
115  }).then(function(allowance) {
116    assert.equal(allowance.toNumber(), 0, 'deducts the amount from the allowance');
117  });
118 });
119 });
120
```

```
Administrator: Windows Command Processor
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...

Compilation warnings encountered:

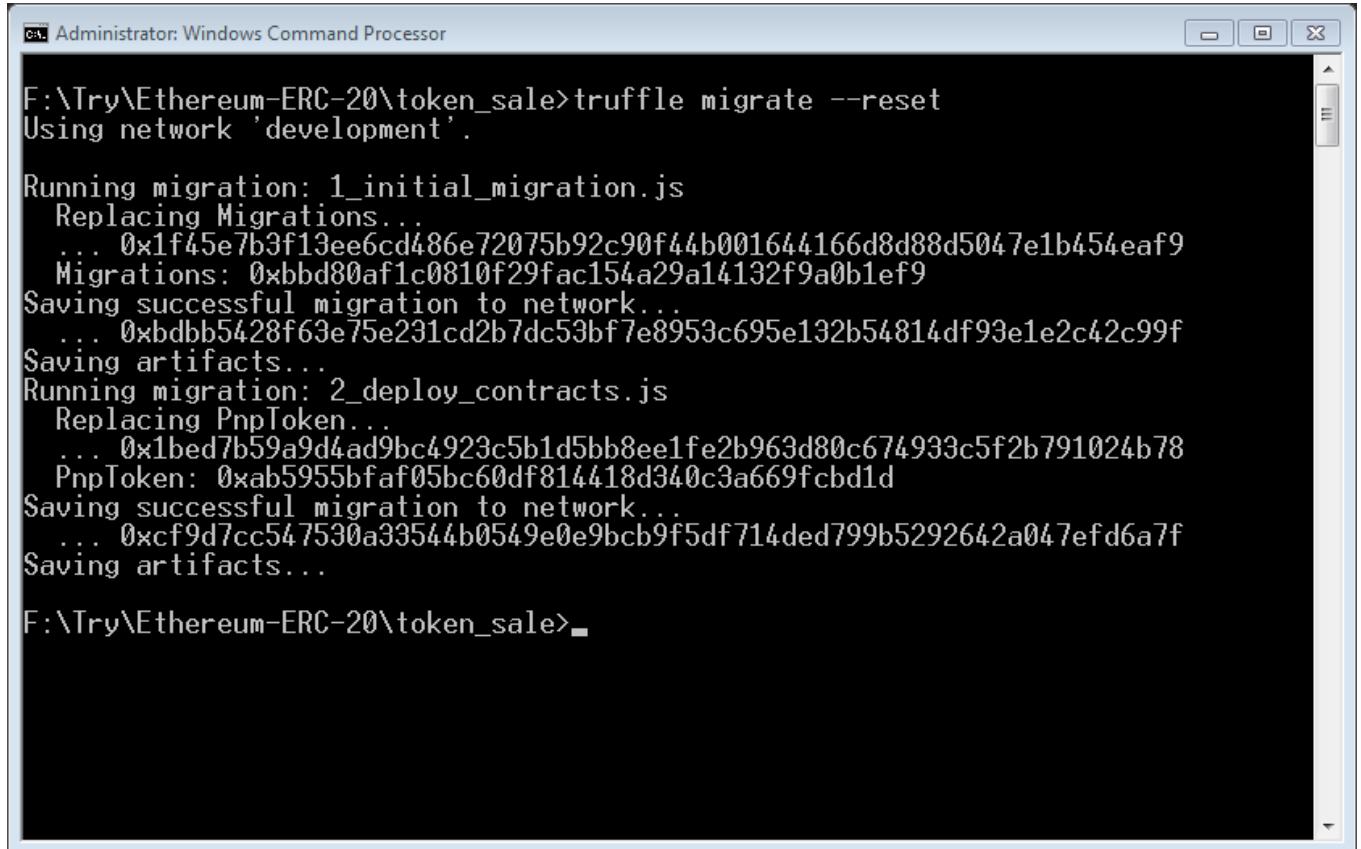
/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:22:2: Warning: Defini
constructors as functions with the same name as the contract is deprecated. U
"constructor(...){ ... }" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:31:2: Warning: Invok
g events without "emit" prefix is deprecated.
    Transfer(msg.sender, _to, _value); // Transfer Event
^-----^
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:37:3: Warning: Invok
g events without "emit" prefix is deprecated.
    Approval(msg.sender,_spender,_value); // Approve Event
^-----^
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:47:9: Warning: Invok
g events without "emit" prefix is deprecated.
    Transfer(_from, _to, _value);
^-----^

Contract: PnpToken
  ✓ initializes the contract with correct values (47ms)
  ✓ allocates the initial supply upon deployment
  ✓ transfers token ownership (124ms)
  ✓ approves tokens for delegated transfer (71ms)
  ✓ handles delegated token transfers (396ms)

5 passing (778ms)
```

Session 7: Console Demonstration

1. Reset Truffle

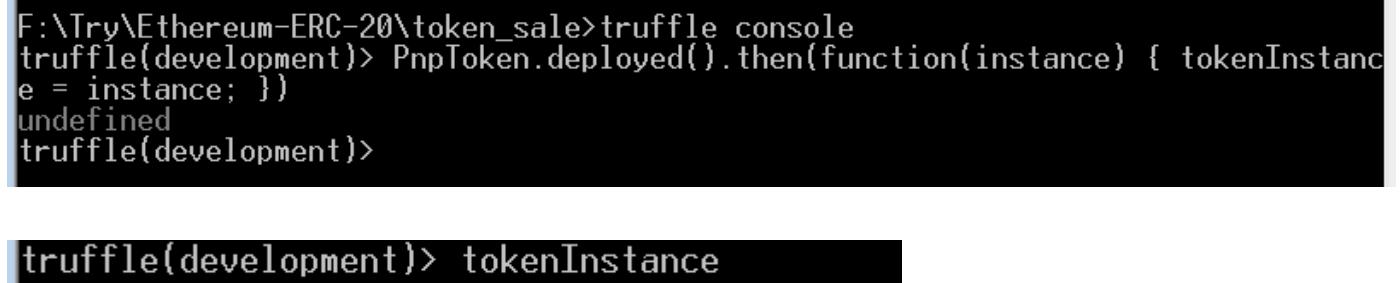


```
Administrator: Windows Command Processor
F:\Try\Ethereum-ERC-20\token_sale>truffle migrate --reset
Using network 'development'.

Running migration: 1_initial_migration.js
  Replacing Migrations...
    ... 0x1f45e7b3f13ee6cd486e72075b92c90f44b001644166d8d88d5047e1b454eaf9
      Migrations: 0xbbd80af1c0810f29fac154a29a14132f9a0b1ef9
Saving successful migration to network...
    ... 0xbdbb5428f63e75e231cd2b7dc53bf7e8953c695e132b54814df93e1e2c42c99f
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Replacing PnpToken...
    ... 0x1bed7b59a9d4ad9bc4923c5b1d5bb8ee1fe2b963d80c674933c5f2b791024b78
      PnpToken: 0xab5955bfaf05bc60df814418d340c3a669fcbd1d
Saving successful migration to network...
    ... 0xcf9d7cc547530a33544b0549e0e9bcb9f5df714ded799b5292642a047efd6a7f
Saving artifacts...

F:\Try\Ethereum-ERC-20\token_sale>
```

2. Go to console



```
F:\Try\Ethereum-ERC-20\token_sale>truffle console
truffle(development)> PnpToken.deployed().then(function(instance) { tokenInstance = instance; })
undefined
truffle(development)>
```

truffle(development)> tokenInstance

```
Administrator: Windows Command Processor - truffle console
{ [Function]
  call: [Function],
  sendTransaction: [Function],
  request: [Function: bound ],
  estimateGas: [Function] },
allowance:
{ [Function]
  call: [Function],
  sendTransaction: [Function],
  request: [Function: bound ],
  estimateGas: [Function] },
Transfer: { [Function: bound ] 'address,address,uint256': [Function: bound ] }
Approval: { [Function: bound ] 'address,address,uint256': [Function: bound ] }
transfer:
{ [Function]
  call: [Function],
  sendTransaction: [Function],
  request: [Function: bound ],
  estimateGas: [Function] },
approve:
{ [Function]
  call: [Function],
  sendTransaction: [Function],
  request: [Function: bound ],
  estimateGas: [Function] },
transferFrom:
{ [Function]
  call: [Function],
  sendTransaction: [Function],
  request: [Function: bound ],
  estimateGas: [Function] },
sendTransaction: [Function],
send: [Function],
allEvents: [Function: bound ],
address: '0xab5955bfaf05bc60df814418d340c3a669fcfd1d',
transactionHash: null }
truffle(development)>
```

```
truffle(development)> tokenInstance.name();
'PNP Token'
truffle(development)> tokenInstance.symbol();
'PNP'
truffle(development)> tokenInstance.standard();
'PNP Token v1.0'
truffle(development)> tokenInstance.totalSupply();
BigNumber { s: 1, e: 6, c: [ 1000000 ] }
truffle(development)> tokenInstance.totalSupply().then(function(s) { supply =s;})
undefined
truffle(development)> supply
BigNumber { s: 1, e: 6, c: [ 1000000 ] }
truffle(development)> supply.toNumber()
1000000
```

```
truffle(development)> web3.eth.accounts()
TypeError: web3.eth.accounts is not a function
truffle(development)> web3.eth.accounts();
TypeError: web3.eth.accounts is not a function
truffle(development)> web3.eth.accounts
[ '0xb6d6510c99c4387c6d51c3d36642b8ce5269f518',
  '0x0f453b50beccdd51dad5966d07132e53d20d25dd',
  '0x68b090fcbb349f24041b5d68dae173936f418251',
  '0x7e4be9b7d977f6717ad8d953933f0806b1cfee58',
  '0xd56ce63ce7fe32de29a2e472c27152d542aca558',
  '0x5695118d159c742d5aefbcba546ffd1801bcd945',
  '0x2b47b32e48067ab399055cd0b0460fa5124f3f69',
  '0x8e3610c3b588a738eecb42866b7ac1cc239adce6',
  '0x25d6713b3666e2406db12c52d7aec78ee7055ee',
  '0x383b685474681142ace9fc0037b042e317f6cd1f' ]
truffle(development)> web3.eth.accounts[0]
'0xb6d6510c99c4387c6d51c3d36642b8ce5269f518'
truffle(development)> admin = web3.eth.accounts[0]
'0xb6d6510c99c4387c6d51c3d36642b8ce5269f518'
truffle(development)> tokenInstance.balanceOf(admin)
BigNumber { s: 1, e: 6, c: [ 1000000 ] }
truffle(development)> tokenInstance.balanceOf(admin).then(function(bal) { balance = bal; })
undefined
truffle(development)> balance.to
balance.toLocaleString balance.toString

balance.toDigits      balance.toExponential  balance.toFixed
balance.toFormat      balance.toFraction    balance.toJSON
balance.toNumber     balance.toPower       balance.toPrecision

truffle(development)> balance.toNumber()
1000000
truffle(development)>
```



```
truffle(development)> tokenInstance.balanceOf(fromAccount)
BigNumber { s: 1, e: 1, c: [ 90 ] }
truffle(development)> tokenInstance.balanceOf(toAccount)
BigNumber { s: 1, e: 1, c: [ 10 ] }
```

```
truffle(development)> tokenInstance.allowance(fromAccount, spendingAccount)
BigNumber { s: 1, e: 0, c: [ 0 ] }
truffle(development)>
```

Session 8: Creating the Crowd Sale Smart Contract

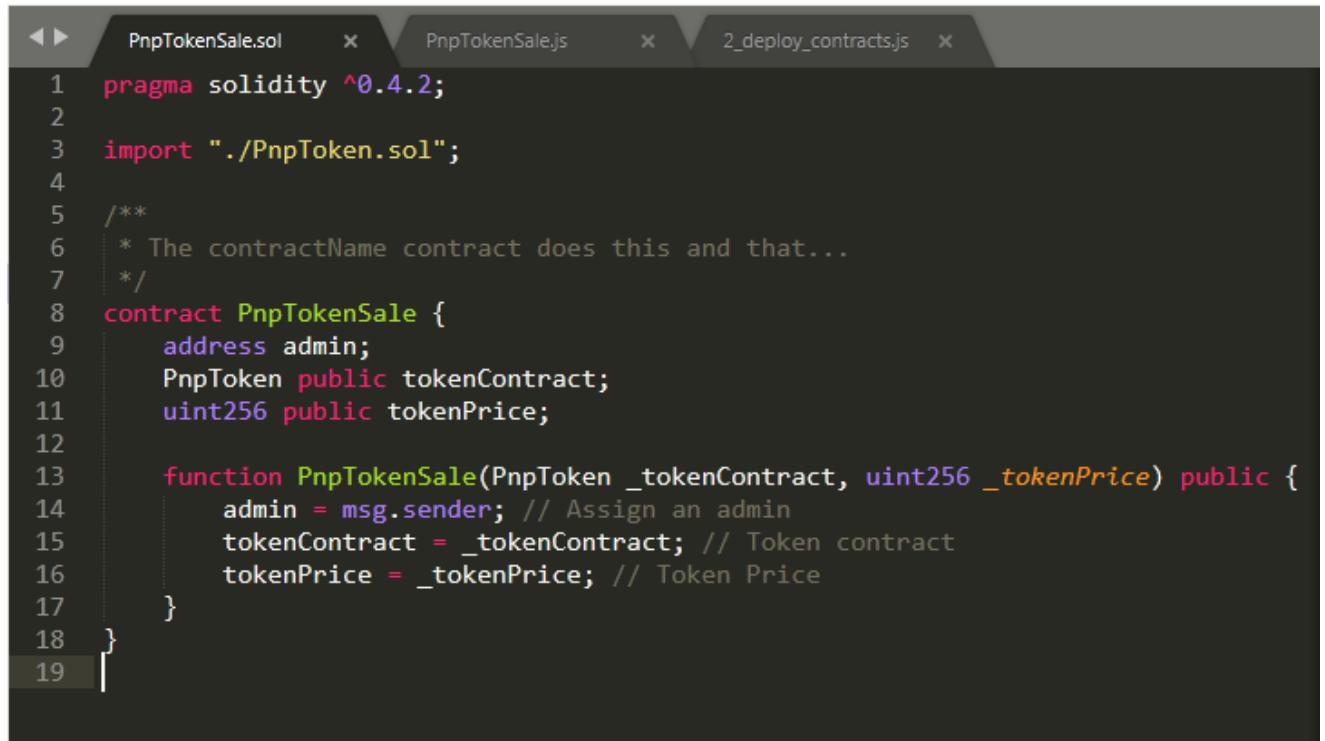
1. Reset Truffle

```
F:\Try\Ethereum-ERC-20\token_sale>truffle migrate --reset
Using network 'development'.

Running migration: 1_initial_migration.js
  Replacing Migrations...
    ... 0x103b5d5881fc29f07aa71d8e8bcefde40930c2d6c4b531fa8e4b4aedddd7b54
    Migrations: 0x3897414d06b5c81aa306f3397a2c4b604562587e
Saving successful migration to network...
    ... 0x5192b05f86808a026da954fddef10e641bab116edbfcf33155dbb5196f5b6ed6
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Replacing PnpToken...
    ... 0x6459d1d8522801fc352b5f96f5e51e3aca63cb2251303d7c8f609109ec621fa
    PnpToken: 0xb57f8744dfa94f43c677fd4fdc6a5ff9a935bc98
Saving successful migration to network...
    ... 0x9ebd1abe7e95cf3ee4b29a71cf8f98fa8150325716a2bc658fdaf161eda73859
Saving artifacts...

F:\Try\Ethereum-ERC-20\token_sale>
```

2. Create a files “PnpTokenSale.sol” in token_sale\contracts , “PnpTokenSale.js” in token_sale\test and follow the code as below



```

1  pragma solidity ^0.4.2;
2
3  import "./PnpToken.sol";
4
5  /**
6   * The contractName contract does this and that...
7   */
8  contract PnpTokenSale {
9      address admin;
10     PnpToken public tokenContract;
11     uint256 public tokenPrice;
12
13     function PnpTokenSale(PnpToken _tokenContract, uint256 _tokenPrice) public {
14         admin = msg.sender; // Assign an admin
15         tokenContract = _tokenContract; // Token contract
16         tokenPrice = _tokenPrice; // Token Price
17     }
18 }
19

```

```
◀ ▶ PnPTokenSale.sol ✘ PnPTokenSale.js ✘ 2_deploy_contracts.js ✘
1 var PnPTokenSale = artifacts.require("./PnPTokenSale.sol");
2
3 contract('PnPTokenSale', function(accounts) {
4     var tokenSaleInstance;
5     var tokenPrice = 1000000000000000; // in wei
6
7
8     it('initializes the contract with the correct values', function() {
9         return PnPTokenSale.deployed().then(function(instance) {
10             tokenSaleInstance = instance;
11             return tokenSaleInstance.address
12         }).then(function(address) {
13             assert.notEqual(address, 0x0, 'has contract address');
14             return tokenSaleInstance.tokenContract();
15         }).then(function(address) {
16             assert.notEqual(address, 0x0, 'has token contract address');
17             return tokenSaleInstance.tokenPrice();
18         }).then(function(price) {
19             assert.equal(price, tokenPrice, 'token price is correct');
20         });
21     });
22
23 });

```

```
◀ ▶ PnPTokenSale.sol ✘ PnPTokenSale.js ✘ 2_deploy_contracts.js ✘
1 var PnPToken = artifacts.require("./PnPToken.sol");
2 var PnPTokenSale = artifacts.require("./PnPTokenSale.sol");
3
4
5 module.exports = function(deployer) {
6     deployer.deploy(PnPToken, 100000).then(function(){
7         // Token price is 0.001 Ether
8         var tokenPrice = 1000000000000000;
9         return deployer.deploy(PnPTokenSale, PnPToken.address, tokenPrice);
10    });
11 };
12
13

```

```
Administrator: Windows Command Processor
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compiling .\contracts\PnpTokenSale.sol...

Compilation warnings encountered:

/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:21:2: Warning: Defining constructors
"as functions with the same name as the contract is deprecated. Use "constructor(...){ ... }"
instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpTokenSale.sol:13:2: Warning: Defining constructors
as functions with the same name as the contract is deprecated. Use "constructor(...){ ... }"
instead.
    function PnpTokenSale(PnpToken _tokenContract, uint256 _tokenPrice) public {
^ (Relevant source part starts here and spans across multiple lines).
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:30:2: Warning: Invoking events without
"emit" prefix is deprecated.
    Transfer(msg.sender, _to, _value); // Transfer Event
^_____
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:36:3: Warning: Invoking events without
"emit" prefix is deprecated.
    Approval(msg.sender, _spender, _value); // Approve Event
^_____
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:46:9: Warning: Invoking events without
"emit" prefix is deprecated.
    Transfer(_from, _to, _value);
^_____

Contract: PnpToken
↳ initializes the contract with correct values (58ms)
↳ allocates the initial supply upon deployment
↳ transfers token ownership (154ms)
↳ approves tokens for delegated transfer (75ms)
↳ handles delegated token transfers (399ms)

Contract: PnpTokenSale
↳ initializes the contract with the correct values

6 passing (860ms)

F:\Try\Ethereum-ERC-20\token_sale>
```

Session 9: Purchasing ERC-20 Tokens

1. Write code for buying tokens

The screenshot shows a code editor with three tabs: `PnpTokenSale.sol`, `PnpTokenSale.js`, and `2_deploy_contracts.js`. The `PnpTokenSale.sol` tab is active, displaying the following Solidity code:

```

1 pragma solidity ^0.4.2;
2
3 import "./PnpToken.sol";
4
5 /**
6  * The contractName contract does this and that...
7 */
8 contract PnpTokenSale {
9     address admin;
10    PnpToken public tokenContract;
11    uint256 public tokenPrice;
12    uint256 public tokensSold;
13
14    event Sell(address _buyer, uint256 _amount);
15
16    function PnpTokenSale(PnpToken _tokenContract, uint256 _tokenPrice) public {
17        admin = msg.sender; // Assign an admin
18        tokenContract = _tokenContract; // Token contract
19        tokenPrice = _tokenPrice; // Token Price
20    }
21    // multiply function
22    function multiply(uint x, uint y) internal pure returns (uint z) {
23        require(y == 0 || (z = x * y) / y == x);
24    }
25    // Buy Tokens
26    function buyTokens(uint256 _numberOfTokens) public payable {
27        require(msg.value == multiply(_numberOfTokens, tokenPrice)); // Require that value equal to tokens
28        require(tokenContract.balanceOf(this) >= _numberOfTokens); // Require that contract has enough tokens
29        require(tokenContract.transfer(msg.sender, _numberOfTokens)); // Require that a transfer is successful
30
31        tokensSold += _numberOfTokens;
32
33        Sell(msg.sender, _numberOfTokens); // Trigger Sell Event
34    }
35}
36

```

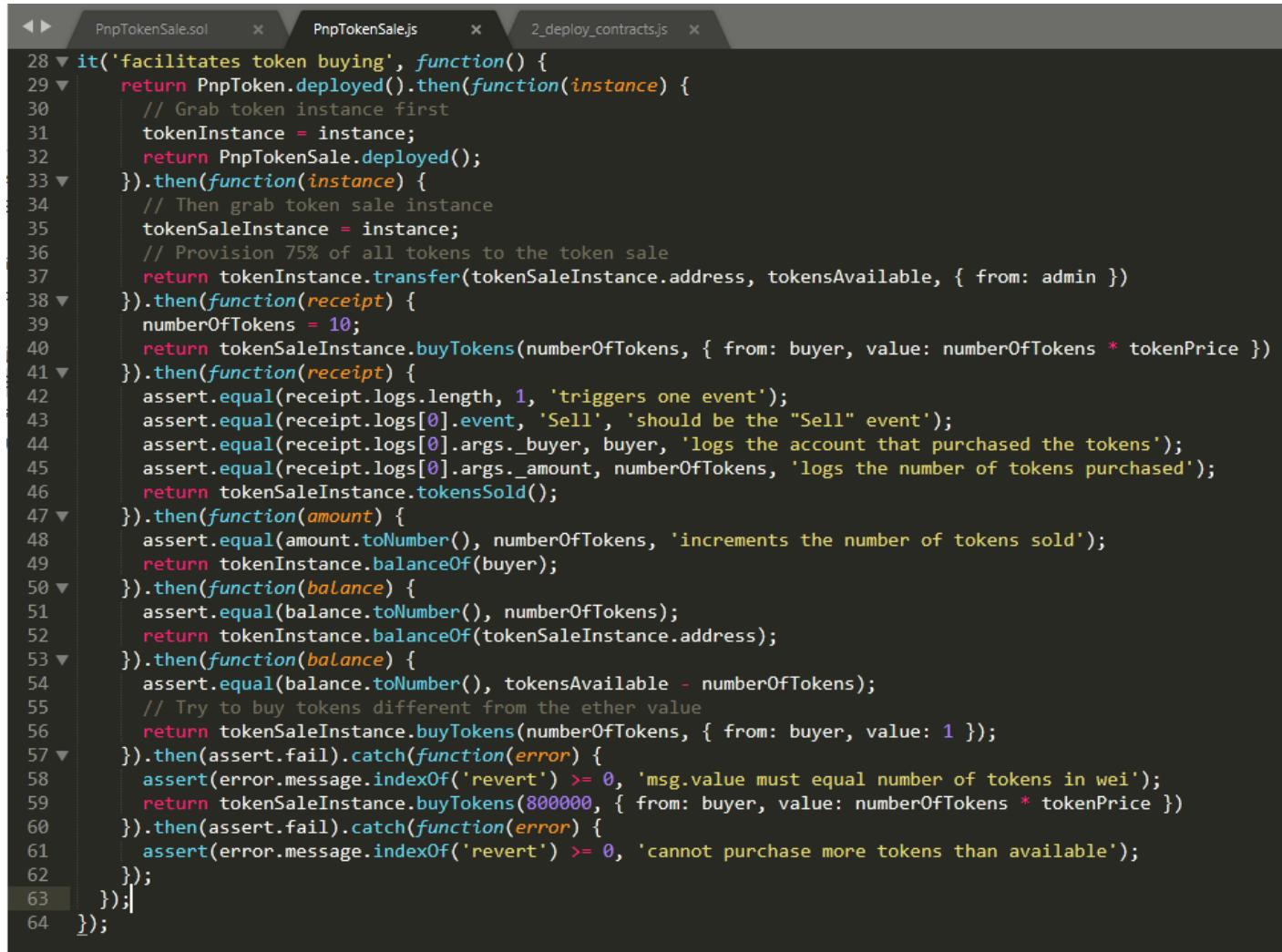
The screenshot shows a code editor with three tabs: `PnpTokenSale.sol`, `PnpTokenSale.js`, and `2_deploy_contracts.js`. The `2_deploy_contracts.js` tab is active, displaying the following JavaScript code:

```

1 var PnpToken = artifacts.require("./PnpToken.sol");
2 var PnpTokenSale = artifacts.require("./PnpTokenSale.sol");
3
4
5 module.exports = function(deployer) {
6     deployer.deploy(PnpToken, 1000000).then(function(){
7         // Token price is 0.001 Ether
8         var tokenPrice = 1000000000000000;
9         return deployer.deploy(PnpTokenSale, PnpToken.address, tokenPrice);
10    });
11 };
12
13

```

```
1 var PnpToken = artifacts.require('./PnpToken.sol');
2 var PnpTokenSale = artifacts.require("./PnpTokenSale.sol");
3
4 ▼ contract('PnpTokenSale', function(accounts) {
5     var tokenInstance;
6     var tokenSaleInstance;
7     var admin = accounts[0];
8     var buyer = accounts[1];
9     var tokenPrice = 1000000000000000; // in wei
10    var tokensAvailable = 750000;
11    var numberofTokens;
12
13    it('initializes the contract with the correct values', function() {
14        return PnpTokenSale.deployed().then(function(instance) {
15            tokenSaleInstance = instance;
16            return tokenSaleInstance.address
17        }).then(function(address) {
18            assert.notEqual(address, 0x0, 'has contract address');
19            return tokenSaleInstance.tokenContract();
20        }).then(function(address) {
21            assert.notEqual(address, 0x0, 'has token contract address');
22            return tokenSaleInstance.tokenPrice();
23        }).then(function(price) {
24            assert.equal(price, tokenPrice, 'token price is correct');
25        });
26    });
27}
```



```
28 ▼ it('facilitates token buying', function() {
29   return PnpToken.deployed().then(function(instance) {
30     // Grab token instance first
31     tokenInstance = instance;
32     return PnpTokenSale.deployed();
33   }).then(function(instance) {
34     // Then grab token sale instance
35     tokenSaleInstance = instance;
36     // Provision 75% of all tokens to the token sale
37     return tokenInstance.transfer(tokenSaleInstance.address, tokensAvailable, { from: admin })
38   }).then(function(receipt) {
39     numberOfTokens = 10;
40     return tokenSaleInstance.buyTokens(numberOfTokens, { from: buyer, value: numberOfTokens * tokenPrice })
41   }).then(function(receipt) {
42     assert.equal(receipt.logs.length, 1, 'triggers one event');
43     assert.equal(receipt.logs[0].event, 'Sell', 'should be the "Sell" event');
44     assert.equal(receipt.logs[0].args._buyer, buyer, 'logs the account that purchased the tokens');
45     assert.equal(receipt.logs[0].args._amount, numberOfTokens, 'logs the number of tokens purchased');
46     return tokenSaleInstance.tokensSold();
47   }).then(function(amount) {
48     assert.equal(amount.toNumber(), numberOfTokens, 'increments the number of tokens sold');
49     return tokenInstance.balanceOf(buyer);
50   }).then(function(balance) {
51     assert.equal(balance.toNumber(), numberOfTokens);
52     return tokenInstance.balanceOf(tokenSaleInstance.address);
53   }).then(function(balance) {
54     assert.equal(balance.toNumber(), tokensAvailable - numberOfTokens);
55     // Try to buy tokens different from the ether value
56     return tokenSaleInstance.buyTokens(numberOfTokens, { from: buyer, value: 1 });
57   }).then(assert.fail).catch(function(error) {
58     assert(error.message.indexOf('revert') >= 0, 'msg.value must equal number of tokens in wei');
59     return tokenSaleInstance.buyTokens(800000, { from: buyer, value: numberOfTokens * tokenPrice })
60   }).then(assert.fail).catch(function(error) {
61     assert(error.message.indexOf('revert') >= 0, 'cannot purchase more tokens than available');
62   });
63 });
64 );
```

```
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.

Compiling .\contracts\PnpToken.sol...
Compiling .\contracts\PnpTokenSale.sol...

Compilation warnings encountered:

/F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:21:2: Warning: Defining constructors as functions with the same name as the contract is deprecated. Use "constructor(...){...}" instead.
    function PnpToken(uint256 _initialSupply) public {
^ (Relevant source part starts here and spans across multiple lines).
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpTokenSale.sol:16:2: Warning: Defining constructors as functions with the same name as the contract is deprecated. Use "constructor(...){...}" instead.
    function PnpTokenSale(PnpToken _tokenContract, uint256 _tokenPrice) public {
^ (Relevant source part starts here and spans across multiple lines).
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:30:2: Warning: Invoking events without "emit" prefix is deprecated.
    Transfer(msg.sender, _to, _value); // Transfer Event
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:36:3: Warning: Invoking events without "emit" prefix is deprecated.
    Approval(msg.sender, _spender, _value); // Approve Event
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpToken.sol:46:9: Warning: Invoking events without "emit" prefix is deprecated.
    Transfer(_from, _to, _value);
./F/Try/Ethereum-ERC-20/token_sale/contracts/PnpTokenSale.sol:35:9: Warning: Invoking events without "emit" prefix is deprecated.
    Sell(msg.sender, _numberOfTokens); // Trigger Sell Event

Contract: PnpToken
    ↓ initializes the contract with correct values (54ms)
    ↓ allocates the initial supply upon deployment
    ↓ transfers token ownership (157ms)
    ↓ approves tokens for delegated transfer (70ms)
    ↓ handles delegated token transfers (389ms)

Contract: PnpTokenSale
    ↓ initializes the contract with the correct values
    ↓ facilitates token buying (197ms)

7 passing (1s)
```

Session 10: Ending Token Sale

1. Write code for Ending Token Sale

```
1 pragma solidity ^0.4.2;
2 import "./PnpToken.sol";
3 /**
4  * The contractName contract does this and that...
5  */
6 contract PnpTokenSale {
7     address admin;
8     PnpToken public tokenContract;
9     uint256 public tokenPrice;
10    uint256 public tokensSold;
11
12    event Sell(address _buyer, uint256 _amount);
13
14    function PnpTokenSale(PnpToken _tokenContract, uint256 _tokenPrice) public {
15        admin = msg.sender; // Assign an admin
16        tokenContract = _tokenContract; // Token contract
17        tokenPrice = _tokenPrice; // Token Price
18    }
19    // multiply function
20    function multiply(uint x, uint y) internal pure returns (uint z) {
21        require(y == 0 || (z = x * y) / y == x);
22    }
23    // Buy Tokens
24    function buyTokens(uint256 _numberOfTokens) public payable {
25        require(msg.value == multiply(_numberOfTokens, tokenPrice));
26        // Require that value equal to tokens
27        require(tokenContract.balanceOf(this) >= _numberOfTokens);
28        // Require that contract has enough tokens
29        require(tokenContract.transfer(msg.sender, _numberOfTokens));
30        // Require that a transfer is successful
31        tokensSold += _numberOfTokens;
32        Sell(msg.sender, _numberOfTokens); // Trigger Sell Event
33    }
34    // Ending Token PnpTokenSale
35    function endSale() public {
36        require(msg.sender == admin); // Require admin
37        require(tokenContract.transfer(admin, tokenContract.balanceOf(this)));
38        // Transfer remaining PnpToken to the admin
39        selfdestruct(admin); // Destroy contract
40    }
41 }
42 }
```

```
PnpTokenSale.sol      PnpTokenSale.js      2_deploy_contracts.js
1 var PnpToken = artifacts.require("./PnpToken.sol");
2 var PnpTokenSale = artifacts.require("./PnpTokenSale.sol");
3
4 module.exports = function(deployer) {
5   deployer.deploy(PnpToken, 1000000).then(function(){
6     // Token price is 0.001 Ether
7     var tokenPrice = 1000000000000000;
8     return deployer.deploy(PnpTokenSale, PnpToken.address, tokenPrice);
9   });
10 };
11
12
```

```
PnpTokenSale.sol      PnpTokenSale.js      2_deploy_contracts.js
1 var PnpToken = artifacts.require('./PnpToken.sol');
2 var PnpTokenSale = artifacts.require("./PnpTokenSale.sol");
3
4 contract('PnpTokenSale', function(accounts) {
5   var tokenInstance;
6   var tokenSaleInstance;
7   var admin = accounts[0];
8   var buyer = accounts[1];
9   var tokenPrice = 1000000000000000; // in wei
10  var tokensAvailable = 750000;
11  var numberoftokens;
12
13  it('initializes the contract with the correct values', function() {
14    return PnpTokenSale.deployed().then(function(instance) {
15      tokenSaleInstance = instance;
16      return tokenSaleInstance.address
17    }).then(function(address) {
18      assert.notEqual(address, 0x0, 'has contract address');
19      return tokenSaleInstance.tokenContract();
20    }).then(function(address) {
21      assert.notEqual(address, 0x0, 'has token contract address');
22      return tokenSaleInstance.tokenPrice();
23    }).then(function(price) {
24      assert.equal(price, tokenPrice, 'token price is correct');
25    });
26  });
27
```

```

28  it('facilitates token buying', function() {
29    return PnpToken.deployed().then(function(instance) {
30      // Grab token instance first
31      tokenInstance = instance;
32      return PnpTokenSale.deployed();
33    }).then(function(instance) {
34      // Then grab token sale instance
35      tokenSaleInstance = instance;
36      // Provision 75% of all tokens to the token sale
37      return tokenInstance.transfer(tokenSaleInstance.address, tokensAvailable, { from: admin })
38    }).then(function(receipt) {
39      numberOfTokens = 10;
40      return tokenSaleInstance.buyTokens(numberOfTokens, { from: buyer, value: numberOfTokens * tokenPrice })
41    }).then(function(receipt) {
42      assert.equal(receipt.logs.length, 1, 'triggers one event');
43      assert.equal(receipt.logs[0].event, 'Sell', 'should be the "Sell" event');
44      assert.equal(receipt.logs[0].args._buyer, buyer, 'logs the account that purchased the tokens');
45      assert.equal(receipt.logs[0].args._amount, numberOfTokens, 'logs the number of tokens purchased');
46      return tokenSaleInstance.tokensSold();
47    }).then(function(amount) {
48      assert.equal(amount.toNumber(), numberOfTokens, 'increments the number of tokens sold');
49      return tokenInstance.balanceOf(buyer);
50    }).then(function(balance) {
51      assert.equal(balance.toNumber(), numberOfTokens);
52      return tokenInstance.balanceOf(tokenSaleInstance.address);
53    }).then(function(balance) {
54      assert.equal(balance.toNumber(), tokensAvailable - numberOfTokens);
55      // Try to buy tokens different from the ether value
56      return tokenSaleInstance.buyTokens(numberOfTokens, { from: buyer, value: 1 });
57    }).then(assert.fail).catch(function(error) {
58      assert(error.message.indexOf('revert') >= 0, 'msg.value must equal number of tokens in wei');
59      return tokenSaleInstance.buyTokens(800000, { from: buyer, value: numberOfTokens * tokenPrice })
60    }).then(assert.fail).catch(function(error) {
61      assert(error.message.indexOf('revert') >= 0, 'cannot purchase more tokens than available');
62    });
63  });
64

65
66  it('ends token sale', function() {
67    return PnpToken.deployed().then(function(instance) {
68      // Grab token instance first
69      tokenInstance = instance;
70      return PnpTokenSale.deployed();
71    }).then(function(instance) {
72      // Then grab token sale instance
73      tokenSaleInstance = instance;
74      // Try to end sale from account other than the admin
75      return tokenSaleInstance.endSale({ from: buyer });
76    }).then(assert.fail).catch(function(error) {
77      assert(error.message.indexOf('revert' >= 0, 'must be admin to end sale'));
78      // End sale as admin
79      return tokenSaleInstance.endSale({ from: admin });
80    }).then(function(receipt) {
81      return tokenInstance.balanceOf(admin);
82    }).then(function(balance) {
83      assert.equal(balance.toNumber(), 999990, 'returns all unsold dapp tokens to admin');
84      // Check that the contract has no balance
85      balance = web3.eth.getBalance(tokenSaleInstance.address)
86      assert.equal(balance.toNumber(), 0);
87    });
88  });
89

```

```
F:\Try\Ethereum-ERC-20\token_sale>truffle test
Using network 'development'.


Contract: PnpToken
  ✓ initializes the contract with correct values (52ms)
  ✓ allocates the initial supply upon deployment
  ✓ transfers token ownership (155ms)
  ✓ approves tokens for delegated transfer (70ms)
  ✓ handles delegated token transfers (405ms)

Contract: PnpTokenSale
  ✓ initializes the contract with the correct values
  ✓ facilitates token buying (232ms)
  ✓ ends token sale (397ms)

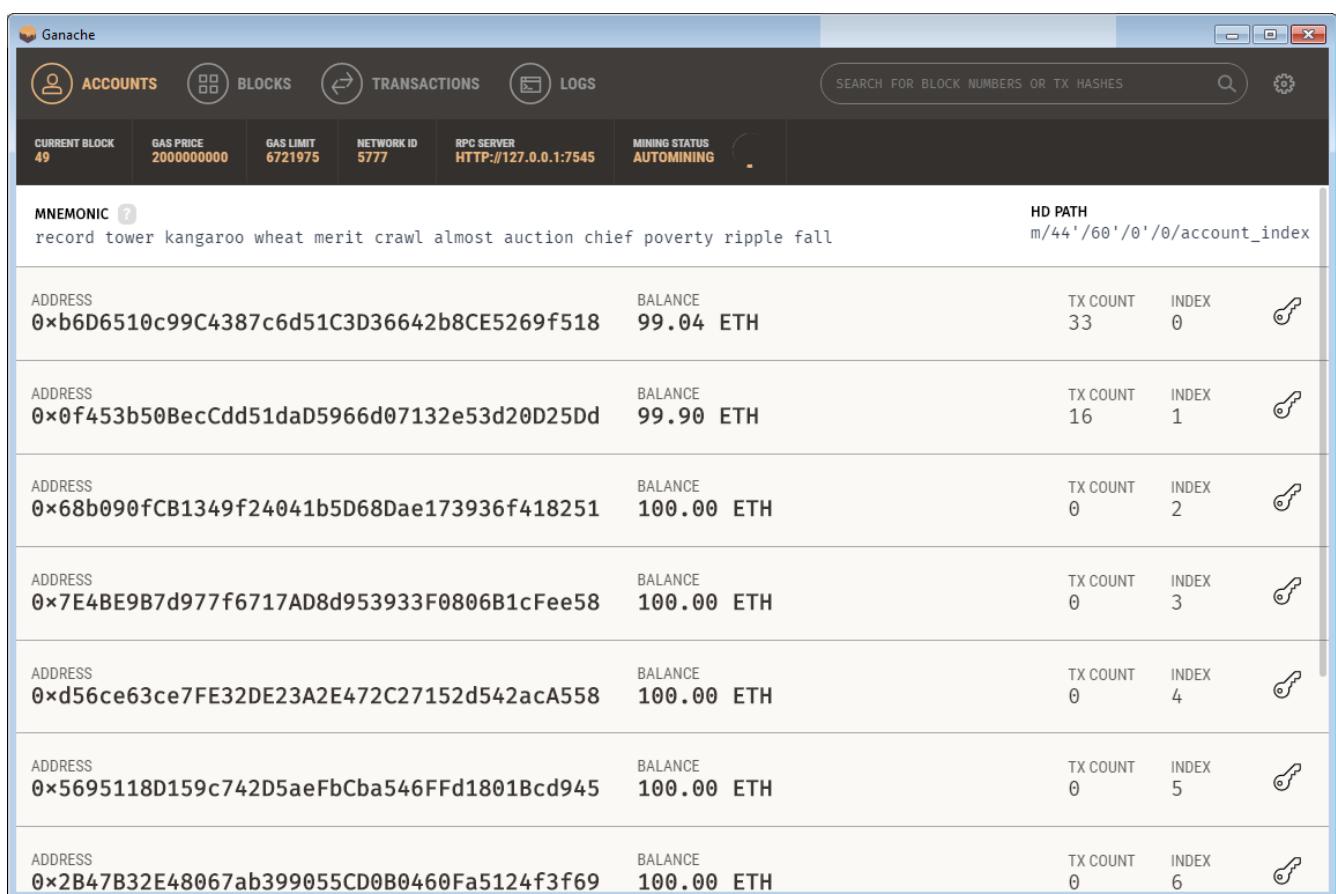
8 passing (2s)

F:\Try\Ethereum-ERC-20\token_sale>
```

Session 11: Creating the Crowd Sale Website

- Check the dependences are installed and working properly or not

```
F:\Try\Ethereum-ERC-20\token_sale>node -v
v8.12.0
```



The screenshot shows the Ganache interface with the following account details:

ADDRESS	BALANCE	TX COUNT	INDEX	
0xb6D6510c99C4387c6d51C3D36642b8CE5269f518	99.04 ETH	33	0	
0x0f453b50BecCdd51daD5966d07132e53d20D25Dd	99.90 ETH	16	1	
0x68b090fCB1349f24041b5D68Dae173936f418251	100.00 ETH	0	2	
0x7E4BE9B7d977f6717AD8d953933F0806B1cFee58	100.00 ETH	0	3	
0xd56ce63ce7FE32DE23A2E472C27152d542acA558	100.00 ETH	0	4	
0x5695118D159c742D5aeFbCba546FFd1801Bcd945	100.00 ETH	0	5	
0x2B47B32E48067ab399055CD0B0460Fa5124f3f69	100.00 ETH	0	6	

- Create “package.json” file and write following code



```

1  {
2    "name": "pnp-token-sale",
3    "version": "1.0.0",
4    "description": "Pnp Token Sale ICO",
5    "main": "truffle.js",
6    "directories": {
7      "test": "test"
8    },
9    "scripts": {
10      "dev": "lite-server"
11    },
12    "author": "Pritesh",
13    "license": "ISC",
14    "devDependencies": {
15      "lite-server": "^2.3.0"
16    }
17 }
```

3. Go to console and write command “npm install”

The screenshot shows two windows. The top window is a 'Administrator: Windows Command Processor' showing the output of the 'npm install' command. The bottom window is a file explorer showing the contents of the 'node_modules' folder.

```
Administrator: Windows Command Processor
operable program or batch file.

F:\>cd Try/
F:\Try>cd Ethereum-ERC-20
F:\Try\Ethereum-ERC-20>cd token_sale
F:\Try\Ethereum-ERC-20\token_sale>node -v
v8.12.0

F:\Try\Ethereum-ERC-20\token_sale>npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN pnp-token-sale@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

added 342 packages from 273 contributors and audited 2609 packages in 32.447s
found 0 vulnerabilities

F:\Try\Ethereum-ERC-20\token_sale>
```

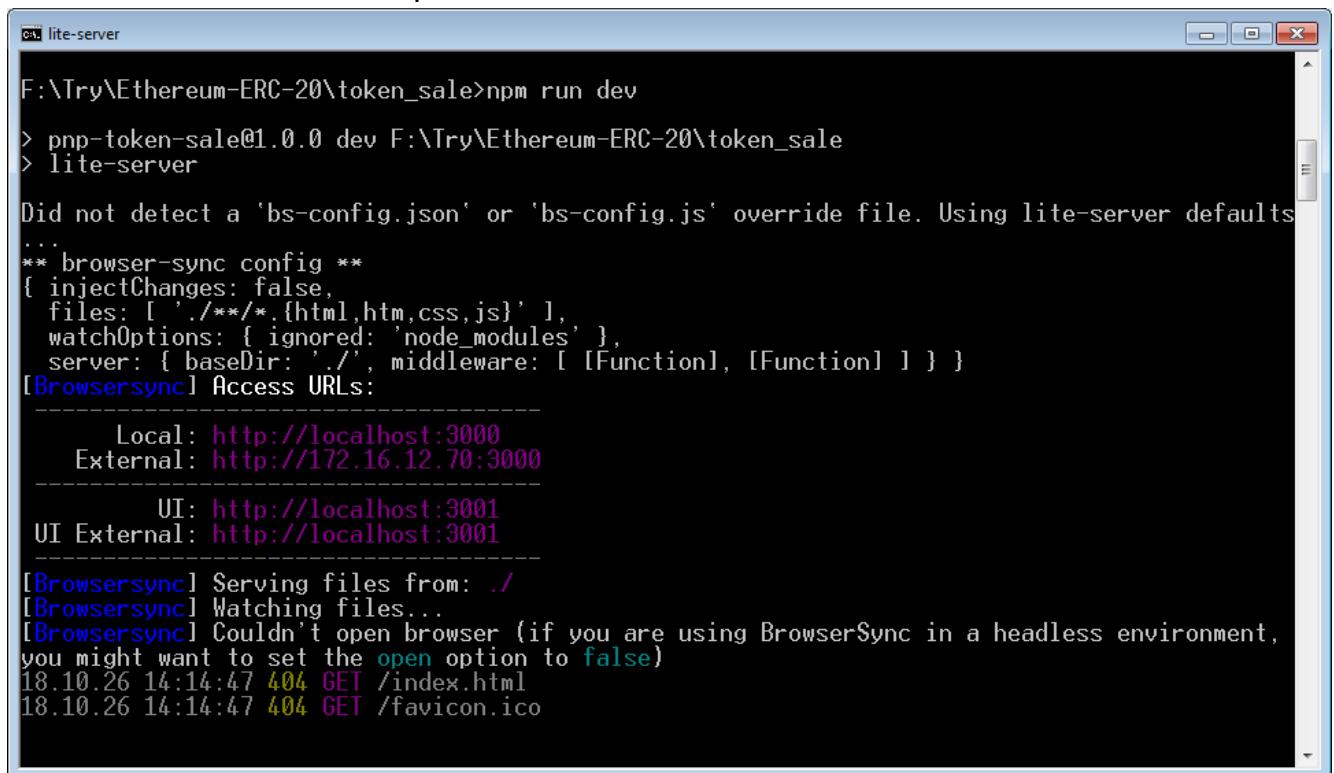
F:\Try\Ethereum-ERC-20\token_sale

File Edit Selection Find View

FOLDERS

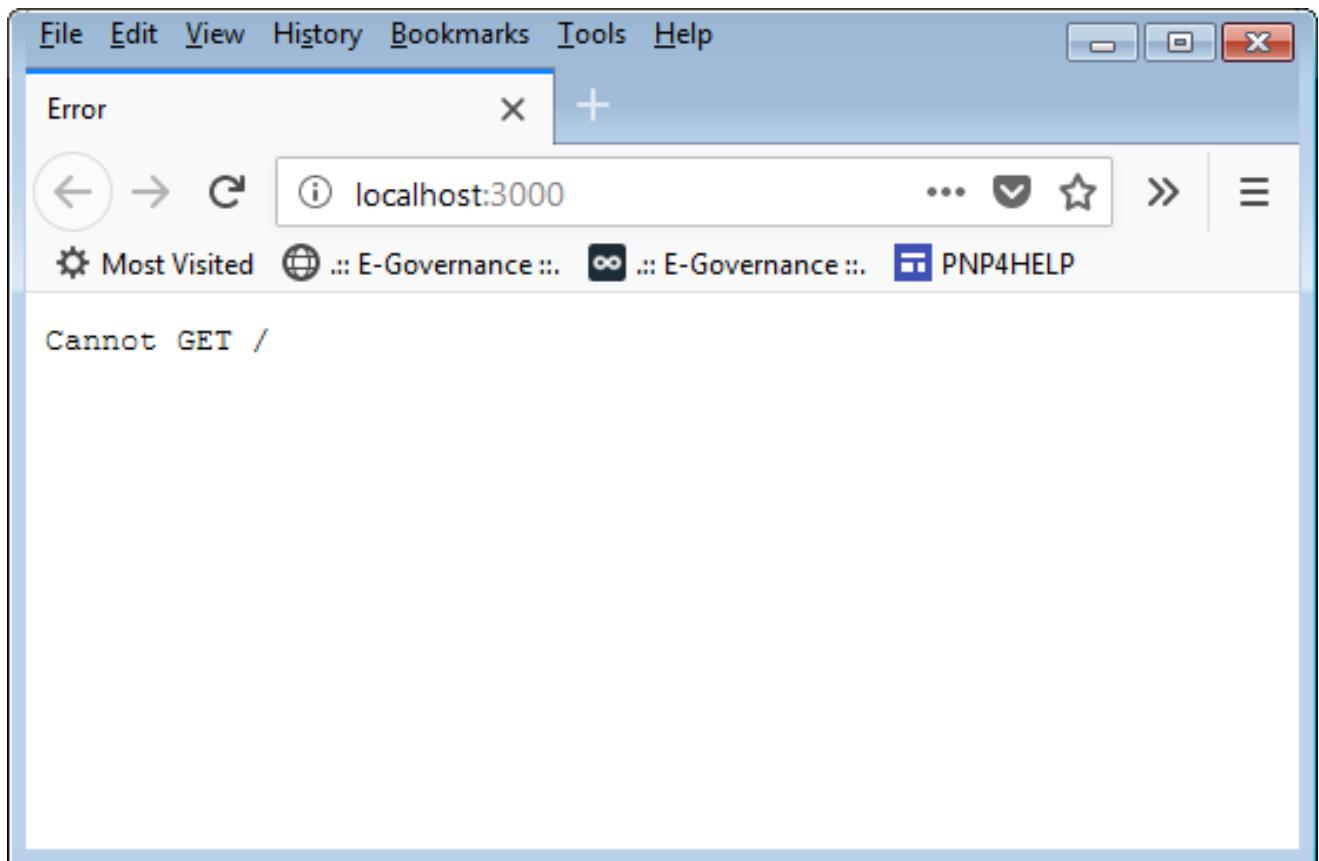
- token_sale
 - build
 - contracts
 - Migrations.sol
 - PnpToken.sol
 - PnpTokenSale.sol
 - migrations
 - node_modules
 - .bin
 - accepts
 - after
 - ansi-regex
 - ansi-styles
 - anymatch
 - arr-diff
 - arr-flatten
 - arr-union
 - array-unique
 - arraybuffer.slice
 - assign-symbols
 - async
 - async-each
 - async-each-series
 - async-limiter
 - atob
 - axios
 - backo2

4. Go to console and write “npm run dev”

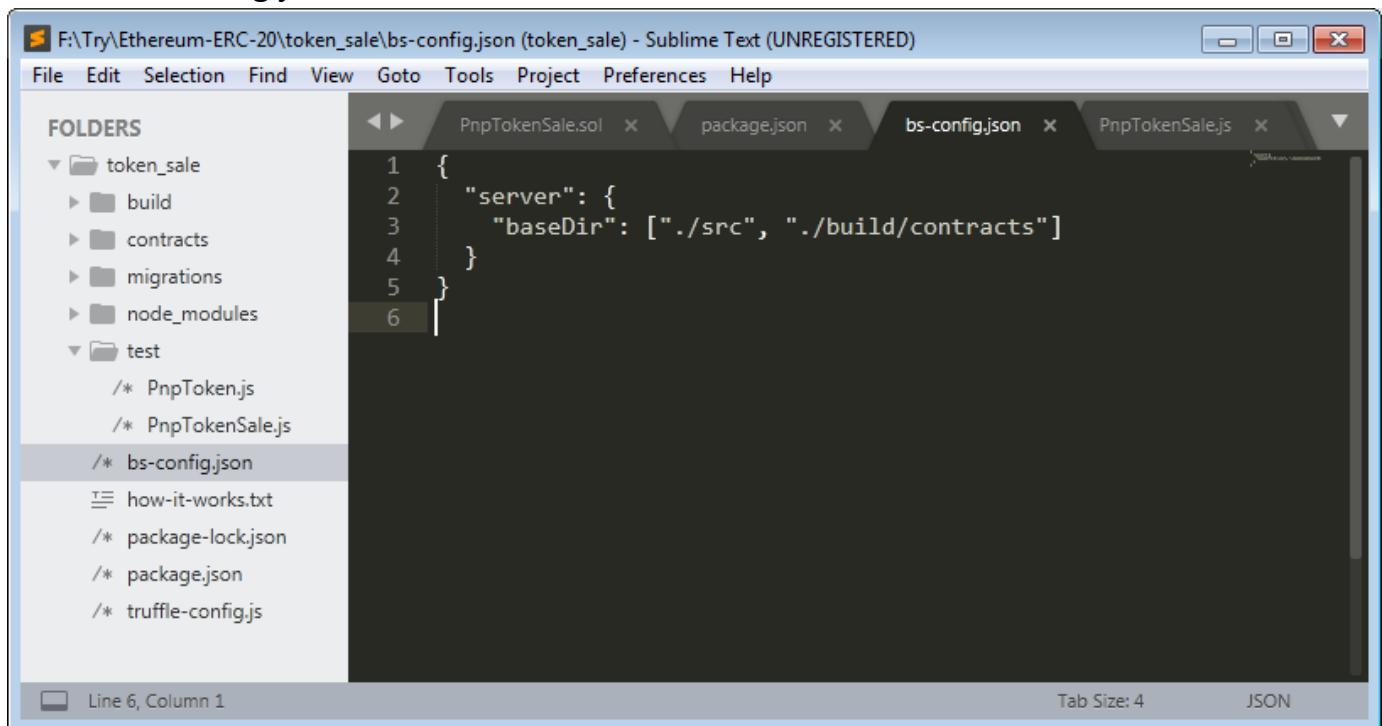


```
F:\Try\Ethereum-ERC-20\token_sale>npm run dev
> pnp-token-sale@1.0.0 dev F:\Try\Ethereum-ERC-20\token_sale
> lite-server

Did not detect a 'bs-config.json' or 'bs-config.js' override file. Using lite-server defaults
...
** browser-sync config **
{
  injectChanges: false,
  files: [ './**/*.{html,htm,css,js}' ],
  watchOptions: { ignored: 'node_modules' },
  server: { baseDir: '.', middleware: [ [Function], [Function] ] }
}
[Browsersync] Access URLs:
-----
   Local: http://localhost:3000
   External: http://172.16.12.70:3000
-----
      UI: http://localhost:3001
UI External: http://localhost:3001
-----
[Browsersync] Serving files from: './'
[Browsersync] Watching files...
[Browsersync] Couldn't open browser (if you are using BrowserSync in a headless environment,
you might want to set the open option to false)
18.10.26 14:14:47 404 GET /index.html
18.10.26 14:14:47 404 GET /favicon.ico
```



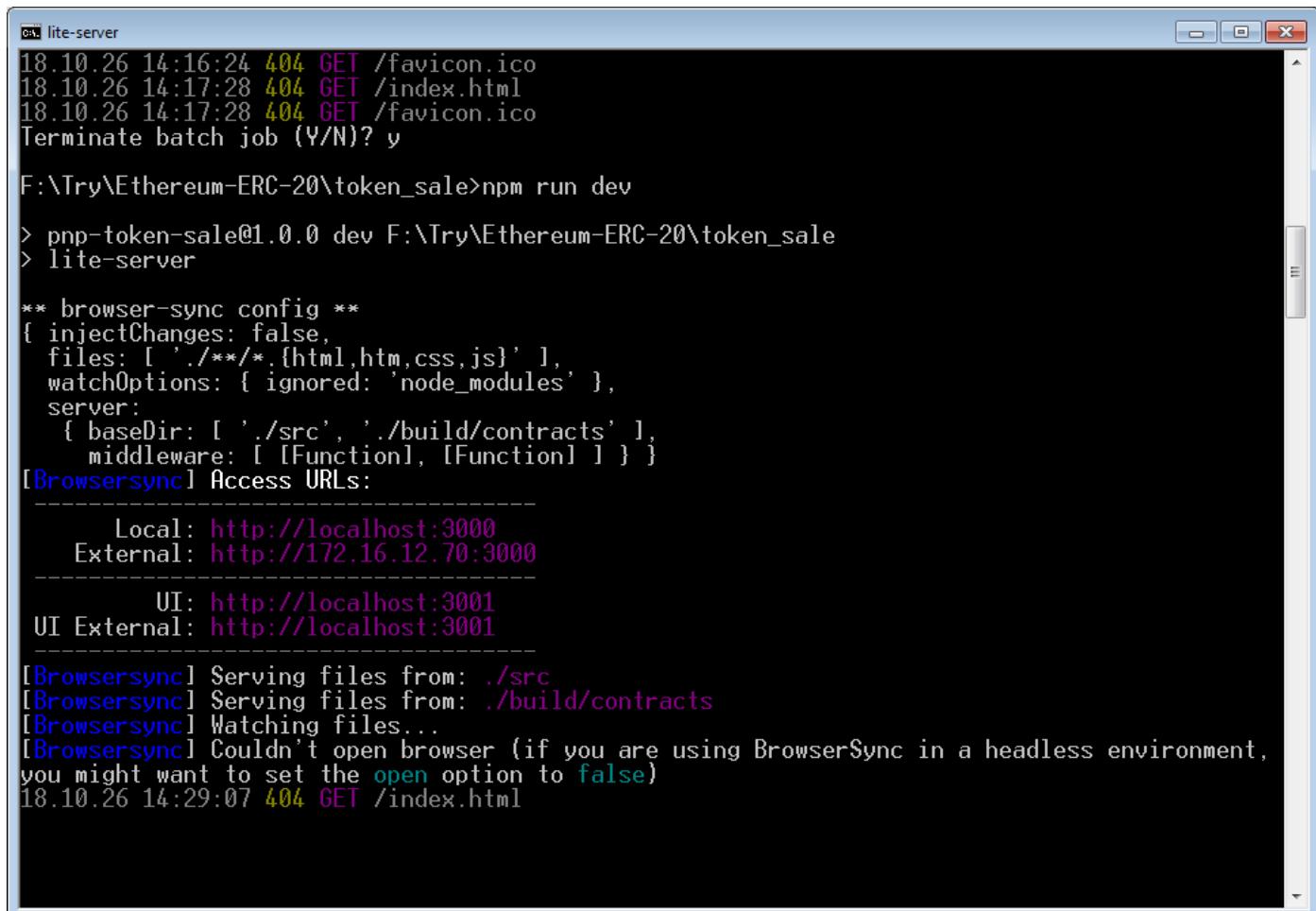
5. Create “bs-config.json” and write below code



```

F:\Try\Ethereum-ERC-20\token_sale\bs-config.json (token_sale) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
token_sale
  build
  contracts
  migrations
  node_modules
  test
    /* PnpToken.js
    /* PnpTokenSale.js
    /* bs-config.json
    how-it-works.txt
    /* package-lock.json
    /* package.json
    /* truffle-config.js
1  {
2    "server": {
3      "baseDir": ["./src", "./build/contracts"]
4    }
5  }
6
Line 6, Column 1
Tab Size: 4
JSON

```



```

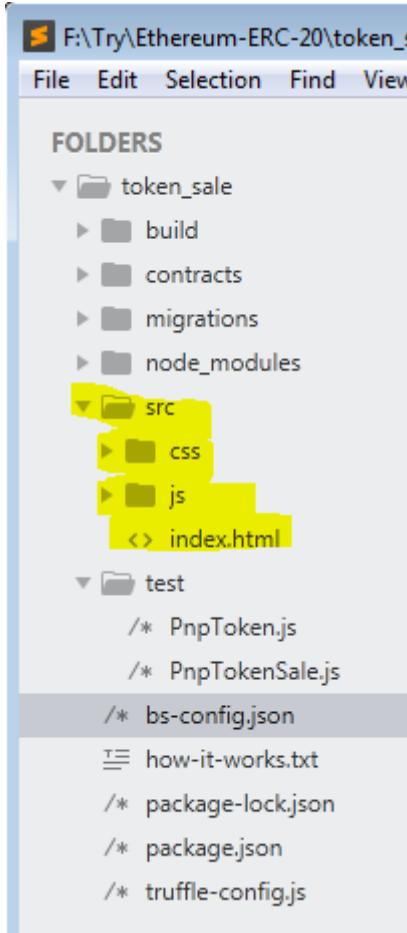
lite-server
18.10.26 14:16:24 404 GET /favicon.ico
18.10.26 14:17:28 404 GET /index.html
18.10.26 14:17:28 404 GET /favicon.ico
Terminate batch job (Y/N)? y

F:\Try\Ethereum-ERC-20\token_sale>npm run dev
> pnp-token-sale@1.0.0 dev F:\Try\Ethereum-ERC-20\token_sale
> lite-server

** browser-sync config **
{
  injectChanges: false,
  files: [ './**/*.{html,htm,css,js}' ],
  watchOptions: { ignored: 'node_modules' },
  server: {
    baseDir: [ './src', './build/contracts' ],
    middleware: [ [Function], [Function] ] }
}
[Browsersync] Access URLs:
-----
  Local: http://localhost:3000
  External: http://172.16.12.70:3000
-----
  UI: http://localhost:3001
  UI External: http://localhost:3001
-----
[Browsersync] Serving files from: ./src
[Browsersync] Serving files from: ./build/contracts
[Browsersync] Watching files...
[Browsersync] Couldn't open browser (if you are using BrowserSync in a headless environment, you might want to set the open option to false)
18.10.26 14:29:07 404 GET /index.html

```

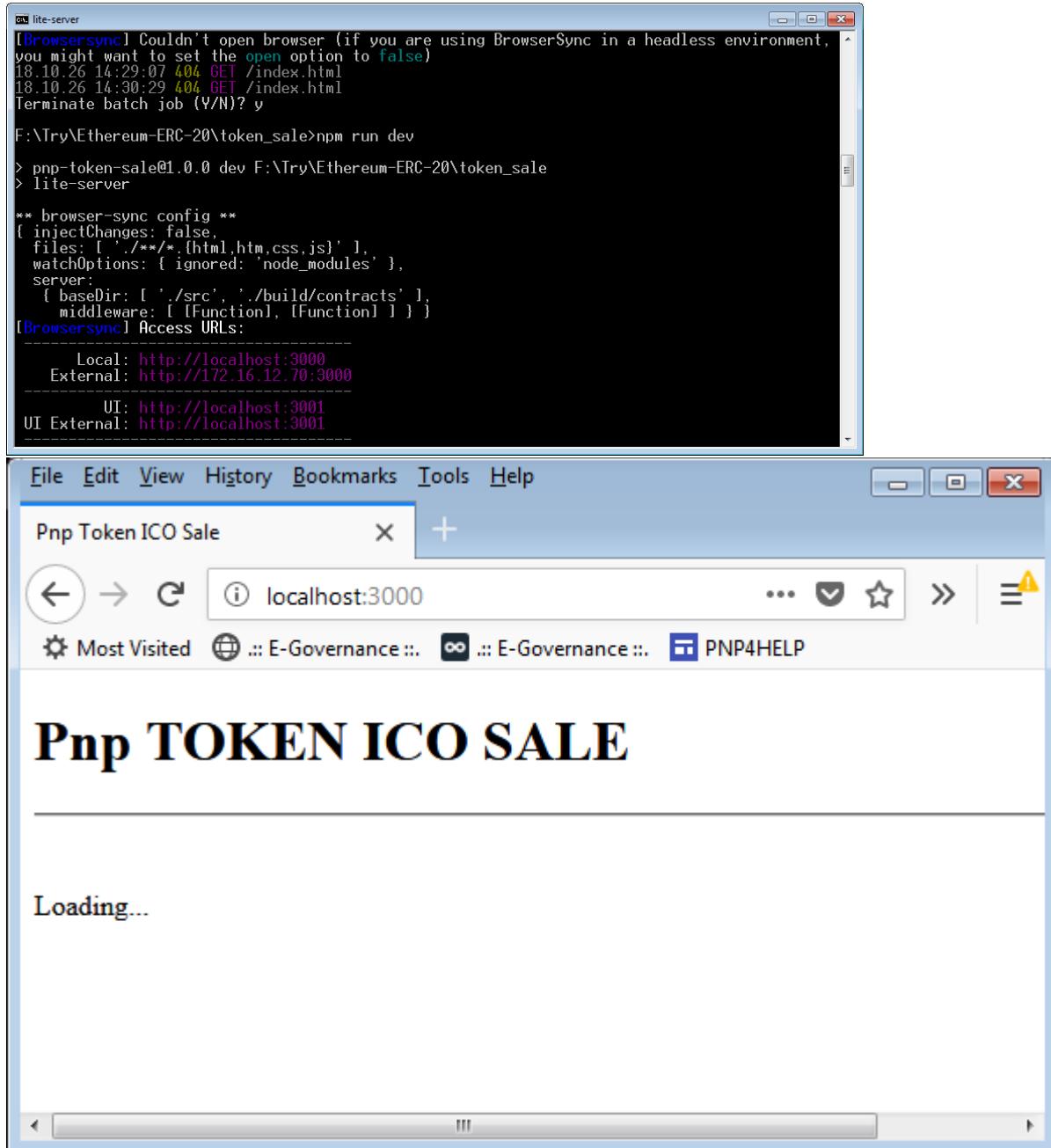
6. Create folders as per following



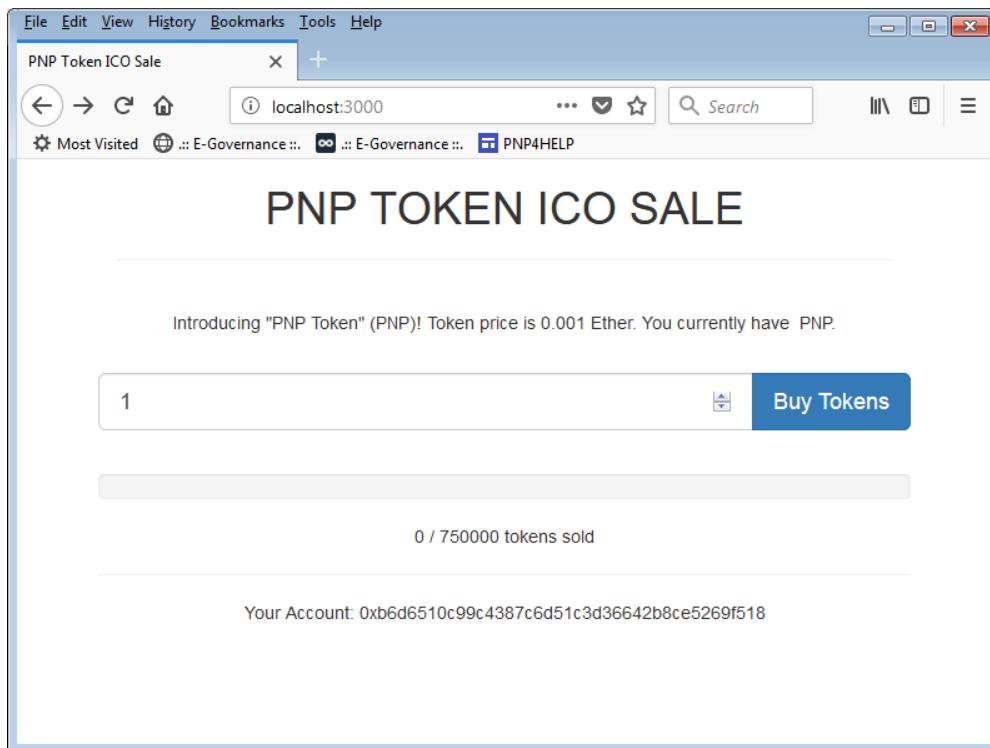
7. Write following code in index.html and go to console then write npm run dev

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>PNP Token ICO Sale</title>
    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <div class="container" style="width: 650px;">
      <div class="row">
        <div class="col-lg-12">
          <h1 class="text-center">PNP TOKEN ICO SALE</h1>
          <hr/>
          <br/>
        </div>
      </div>
    </div>
  </body>
</html>
```

```
</div>
<div id="loader">
  <p class="text-center">Loading...</p>
</div>
<div id="content" class="text-center" style="display: none;">
  <p>
    Introducing "PNP Token" (PNP)!
    Token price is <span class="token-price"></span> Ether. You currently have <span
    class="pnp-balance"></span>&nbsp;PNP.
  </p>
  <br/>
  <form onSubmit="App.buyTokens(); return false;" role="form">
    <div class="form-group">
      <div class="input-group">
        <input id="numberOfTokens" class="form-control input-lg" type="number"
        name="number" value="1" min="1" pattern="[0-9]">
      </input>
      <span class="input-group-btn">
        <button type="submit" class="btn btn-primary btn-lg">Buy Tokens</button>
      </span>
    </div>
    </div>
  </form>
  <br>
  <div class="progress">
    <div id="progress" class="progress-bar progress-bar-striped active" aria-valuemin="0"
    aria-valuemax="100">
      </div>
    </div>
    <p><span class="tokens-sold"></span> / <span class="tokens-available"></span> tokens
    sold</p>
    <hr>
    <p id="accountAddress"></p>
  </div>
  </div>
<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/web3.min.js"></script>
<script src="js/truffle-contract.min.js"></script>
<script src="js/app.js"></script>
</body>
</html>
```



8. Go to https://github.com/dappuniversity/token_sale/tree/master/src/js https://github.com/dappuniversity/token_sale/tree/master/src/css and copy all files



Session 12: Building out the Crowd Sale Website

1. Create app.js in “src/js/” and write below code

```
App = {  
    web3Provider: null,  
    contracts: {},  
    account: '0x0',  
    loading: false,  
    tokenPrice: 10000000000000000,  
    tokensSold: 0,  
    tokensAvailable: 750000,  
  
    init: function() {  
        console.log("App initialized...")  
        return App.initWeb3();  
    },  
  
    initWeb3: function() {  
        if (typeof web3 !== 'undefined') {  
            // If a web3 instance is already provided by Meta Mask.  
            App.web3Provider = web3.currentProvider;  
            web3 = new Web3(web3.currentProvider);  
        } else {  
            // Specify default instance if no web3 instance provided  
            App.web3Provider = new Web3.providers.HttpProvider('http://localhost:7545');  
            web3 = new Web3(App.web3Provider);  
        }  
        return App.initContracts();  
    },  
  
    initContracts: function() {  
        $.getJSON("PnpTokenSale.json", function(pnpTokenSale) {  
            App.contracts.PnpTokenSale = TruffleContract(pnpTokenSale);  
            App.contracts.PnpTokenSale.setProvider(App.web3Provider);  
            App.contracts.PnpTokenSale.deployed().then(function(pnpTokenSale) {  
                console.log("Pnp Token Sale Address:", pnpTokenSale.address);  
            });  
        }).done(function() {  
            $.getJSON("PnpToken.json", function(pnpToken) {  
                App.contracts.PnpToken = TruffleContract(pnpToken);  
                App.contracts.PnpToken.setProvider(App.web3Provider);  
                App.contracts.PnpToken.deployed().then(function(pnpToken) {  
                    console.log("Pnp Token Address:", pnpToken.address);  
                });  
            })  
        })  
        App.listenForEvents();  
    }  
};
```

```
        return App.render();
    });
}
};

// Listen for events emitted from the contract
listenForEvents: function() {
    App.contracts.PnpTokenSale.deployed().then(function(instance) {
        instance.Sell({}, {
            fromBlock: 0,
            toBlock: 'latest',
        }).watch(function(error, event) {
            console.log("event triggered", event);
            App.render();
        })
    });
}

render: function() {
    if (App.loading) {
        return;
    }
    App.loading = true;

    var loader = $('#loader');
    var content = $('#content');

    loader.show();
    content.hide();

    // Load account data
    web3.eth.getCoinbase(function(err, account) {
        if(err === null) {
            App.account = account;
            $('#accountAddress').html("Your Account: " + account);
        }
    });

    // Load token sale contract
    App.contracts.PnpTokenSale.deployed().then(function(instance) {
        pnpTokenSaleInstance = instance;
        return pnpTokenSaleInstance.tokenPrice();
    }).then(function(tokenPrice) {
        App.tokenPrice = tokenPrice;
        $('.token-price').html(web3.fromWei(App.tokenPrice, "ether").toNumber());
        return pnpTokenSaleInstance.tokensSold();
    }).then(function(tokensSold) {
```

```

App.tokensSold = tokensSold.toNumber();
$('.tokens-sold').html(App.tokensSold);
$('.tokens-available').html(App.tokensAvailable);

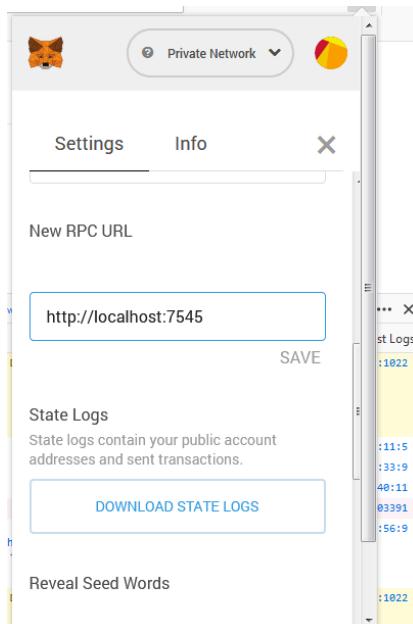
var progressPercent = (Math.ceil(App.tokensSold) / App.tokensAvailable) * 100;
$('#progress').css('width', progressPercent + '%');

// Load token contract
App.contracts.PnpToken.deployed().then(function(instance) {
  pnpTokenInstance = instance;
  return pnpTokenInstance.balanceOf(App.account);
}).then(function(balance) {
  $('.pnp-balance').html(balance.toNumber());
  App.loading = false;
  loader.hide();
  content.show();
})
});
},
buyTokens: function() {
  $('#content').hide();
  $('#loader').show();
  var numberOfTokens = $('#numberOfTokens').val();
  App.contracts.PnpTokenSale.deployed().then(function(instance) {
    return instance.buyTokens(numberOfTokens, {
      from: App.account,
      value: numberOfTokens * App.tokenPrice,
      gas: 500000 // Gas limit
    });
  }).then(function(result) {
    console.log("Tokens bought...")
    $('form').trigger('reset') // reset number of tokens in form
    // Wait for Sell event
  });
}
}

$(function() {
  $(window).load(function() {
    App.init();
  })
});

```

2. Now goto Firefox/Crome Metamask & Select Custom



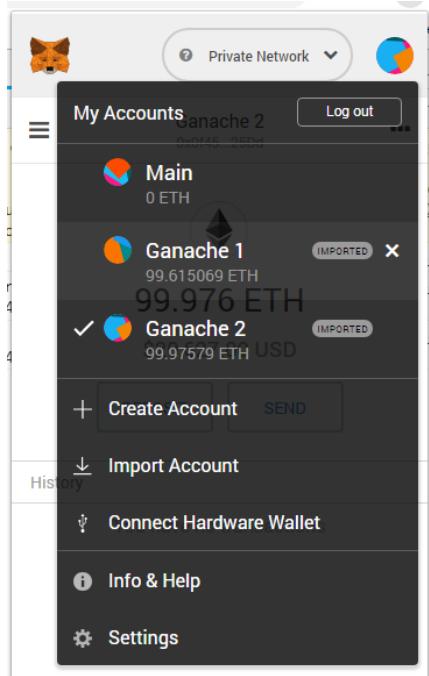
3. Now import Key in Metamask from Ganache (Copy key) & Observe output

The application interface includes a 'Buy Tokens' button and a progress bar indicating '0 / 750000 tokens sold'. Below the progress bar, the account address '0xb6d6510c99c4387c6d51c3d36642b8ce5269f518' is shown.

The Ganache interface displays account details such as 'CURRENT BLOCK 18', 'GAS PRICE 2000000000', 'GAS LIMIT 6721975', 'NETWORK ID 5777', 'RPC SERVER HTTP://127.0.0.1:7545', and 'MINING STATUS AUTOMINING'. It also shows the mnemonic phrase 'record tower kangaroo wheat merit crawl almost auction chief poverty ripple fall' and two account entries with addresses and balances of 99.62 ETH and 99.98 ETH respectively.

Session 13: Purchasing ERC-20 Tokens

1. Import Account 2 From Ganache



2. Go to console and Reset Truffle

```
Administrator: Windows Command Processor
Saving successful migration to network...
... 0x5aeb920b992babab6de2ea2a1032dcbb28ae6102c1a156d58b3277864d6ad6a10
Saving artifacts...

F:\Try\Ethereum-ERC-20\token_sale>truffle migrate --reset
Using network 'development'.

Running migration: 1_initial_migration.js
Replacing Migrations...
... 0xef8e1bc111c687e9830266dc61fe6db3db0e52908e0719406add199b2f184461
Migrations: 0xb422e91e0f9104783abf865f190b0f87c3cccd858
Saving successful migration to network...
... 0x31639fe5f1194070fde7036962d0f1747b32685c8e292ad4e70c3b7f793764a0
Saving artifacts...
Running migration: 2_deploy_contracts.js
Replacing PnpToken...
... 0xccca3bad48290936d5b4aedae02337ecc2867000d6d53eb0ed420c5f4de38b38
PnpToken: 0x80029252fcfd540dbaf5ed5e3862fa4ac34e0ed2
Replacing PnpTokenSale...
... 0xb891beea7f1ddab1d99b76f72154d0a1619e01ef3ec9c8f0fc74182cf6c8b950
PnpTokenSale: 0xb4ae47d878df3e49611fbf7ef8842fc0406e0c1f
Saving successful migration to network...
... 0x0fed7e137f7b54b79eac83df083ca50b2a99c2c996a9d72a0f4c11ed89cbc46
Saving artifacts...

F:\Try\Ethereum-ERC-20\token_sale>
```

3. Go to console and run "npm run dev"

```

lite-server
... 0x15e4d04c71c16d9b22d69c445a8e56c852fa7abc3ca33ee606a0bb92e08c2a39
Saving artifacts...

F:\Try\Ethereum-ERC-20\token_sale>npm run dev
> pnp-token-sale@1.0.0 dev F:\Try\Ethereum-ERC-20\token_sale
> lite-server

** browser-sync config **
{
  injectChanges: false,
  files: [ './**/*.{html,htm,css,js}' ],
  watchOptions: { ignored: 'node_modules' },
  server: {
    baseDir: [ './src', './build/contracts' ],
    middleware: [ [Function], [Function] ] }
}
[Browsersync] Access URLs:
-----
  Local: http://localhost:3000
  External: http://172.16.12.70:3000
-----
  UI: http://localhost:3001
  UI External: http://localhost:3001
-----
[Browsersync] Serving files from: ./src
[Browsersync] Serving files from: ./build/contracts
[Browsersync] Watching files...
[Browsersync] Couldn't open browser (if you are using BrowserSync in a headless environment, you might want to set the open option to false)

```

4. Follow the commands

```

F:\Try\Ethereum-ERC-20\token_sale>truffle console
truffle(development)> PnpTokenSale.deployed().then(function(i) { tokenSale=i; })
undefined
truffle(development)>

Administrator: Windows Command Processor - truffle console
call: [Function],
sendTransaction: [Function],
request: [Function: bound ],
estimateGas: [Function ],
Sell: { [Function: bound ] 'address,uint256': [Function: bound ] },
buyTokens:
{ [Function]
  call: [Function],
  sendTransaction: [Function],
  request: [Function: bound ],
  estimateGas: [Function] },
endSale:
{ [Function]
  call: [Function],
  sendTransaction: [Function],
  request: [Function: bound ],
  estimateGas: [Function] },
sendTransaction: [Function],
send: [Function],
allEvents: [Function: bound ],
address: '0xb4ae47d878df3e49611fbf7ef8842fc0406e0c1f',
transactionHash: null }
truffle(development)> tokenSale
truffle(development)> PnpToken.deployed().then(function(i) { token=i; })
undefined
truffle(development)>

```

Then write token

```
Administrator: Windows Command Processor - truffle console
{
  [Function]
  call: [Function],
  sendTransaction: [Function],
  request: [Function: bound ],
  estimateGas: [Function] },
  approve:
  { [Function]
    call: [Function],
    sendTransaction: [Function],
    request: [Function: bound ],
    estimateGas: [Function] },
  transferFrom:
  { [Function]
    call: [Function],
    sendTransaction: [Function],
    request: [Function: bound ],
    estimateGas: [Function] },
  sendTransaction: [Function],
  send: [Function],
  allEvents: [Function: bound ],
  address: '0x80029252fcfd540dbaf5ed5e9862fa4ac94e0ed2',
  transactionHash: null }
truffle(development)>
```



```
truffle(development)> tokensAvailable = 750000
750000
truffle(development)> tokensAvailable = 750000;
750000
truffle(development)>
```

5. Now Transfer Tokens from one to another account

PNP TOKEN ICO SALE

Introducing "PNP Token" (PNP)! Token price is 0.001 Ether. You currently have 250000 PNP.

1
Buy Tokens

0 / 750000 tokens sold

Your Account: 0x001e54ce86bab90f73e8667e63fb02bda9a83916

Ganache					
ACCOUNTS	BLOCKS	TRANSACTIONS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES	
CURRENT BLOCK 28	GAS PRICE 200000000	GAS LIMIT 6721975	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING
MNEMONIC <small>?</small>					HD PATH m/44'/60'/0'/0/account_index
armed artist bar chimney journey disease mass cruel sense whisper mix latin					
ADDRESS 0x001e54ce86bab90f73e8667e63fb02bda9a83916		BALANCE 98.22 ETH		TX COUNT 24	INDEX 0
ADDRESS 0x7f898ea02D2f8113b92035D78f9073040a545696		BALANCE 100.98 ETH		TX COUNT 4	INDEX 1
ADDRESS 0x74210C2a4B7E3bcbBeB288c6B1B25E50c2E86d27		BALANCE 100.00 ETH		TX COUNT 0	INDEX 2
ADDRESS 0xBe3EB6a8d70F9Ab81775bB132B9FE168Dd96F2CF		BALANCE 100.00 ETH		TX COUNT 0	INDEX 3

The image displays two identical-looking "Send ETH" screens from a Ethereum wallet application, likely MetaMask, running on a "Private Network".

Left Screen (Initial State):

- From:** Ganache 1 (98.222559 ETH / \$20,146.43 USD)
- To:** Recipient Address (Input field)
- Amount:** Main (0x356bC249cA0B408E...) or Max
- Gas Fee:** Ganache 1 (0x001e54cE86bAB90f...) or Ganache 2 (0x7f898ea02D2f8113b...)
- Buttons:** CANCEL (grayed out) and NEXT (light blue)

Right Screen (After Selection):

- From:** Ganache 1 (98.222559 ETH / \$20,146.43 USD)
- To:** 0x7f898ea02D2f8113b9 (Input field)
- Amount:** 50 ETH (\$10,255.50 USD)
- Gas Fee:** 0.000031 ETH (\$0.01 USD)
- Buttons:** CANCEL (grayed out) and NEXT (light blue)

Ethereum-ERC-20

Mr. Priteshkumar Prajapati

The screenshot shows the MetaMask wallet interface. On the left, a transaction confirmation dialog is open, prompting the user to "CONFIRM" a transfer of 50 ETH from Ganache 1 to Ganache 2. The total amount including gas fee is \$10,255.51. On the right, the main wallet screen displays a balance of 48.223 ETH worth \$9,890.44 USD. It shows a history of two recent transactions: one confirmed transfer of 50 ETH and another unconfirmed transfer of 1 ETH.

Transaction Confirmation Dialog:

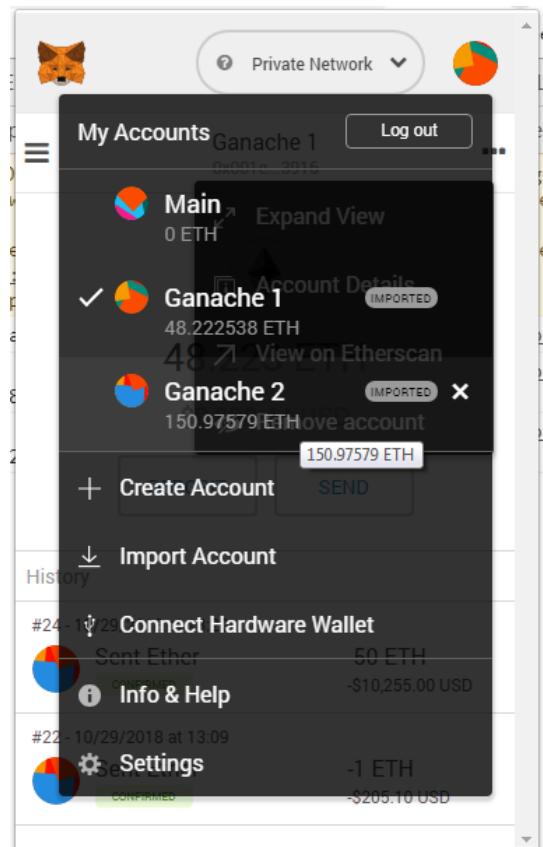
- Amount: 50
- Gas Fee: 0.000031
- Total: 50.000031
- Buttons: REJECT, CONFIRM

Wallet History:

- #24 - 10/29/2018 at 13:20: Sent Ether -50 ETH -\$10,255.00 USD (CONFIRMED)
- #22 - 10/29/2018 at 13:09: Sent Ether -1 ETH -\$205.10 USD (PENDING)

Bottom Confirmation Message:

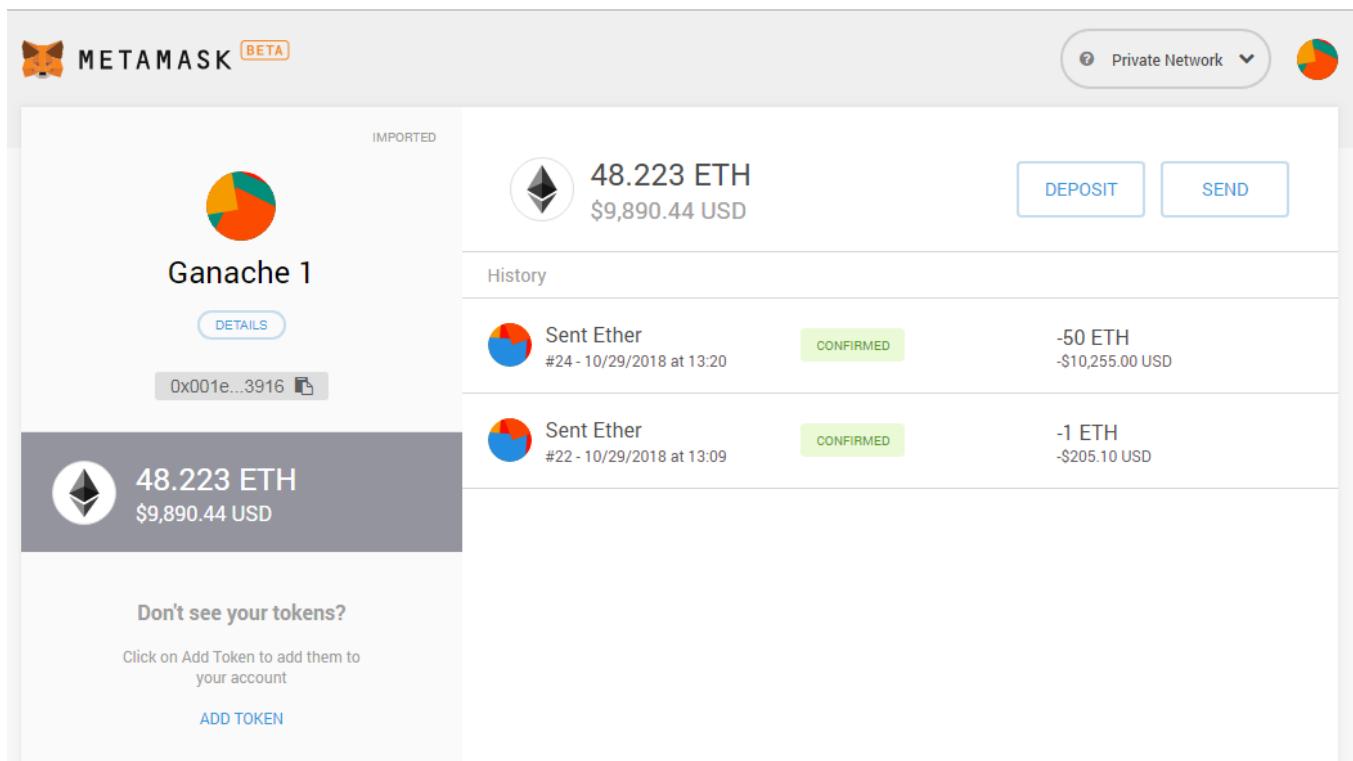
Confirmed transaction
Transaction 24 confirmed! View on EtherScan



The screenshot shows the Ganache UI. At the top, it displays network information: CURRENT BLOCK 29, GAS PRICE 2000000000, GAS LIMIT 6721975, NETWORK ID 5777, RPC SERVER HTTP://127.0.0.1:7545, and MINING STATUS AUTOMINING. A search bar for block numbers or tx hashes is also present.

MNEMONIC: armed artist bar chimney journey disease mass cruel sense whisper mix latin
HD PATH: m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX	
0x001e54cE86bAB90f73E8667E63fb02BDA9A83916	48.22 ETH	25	0	
0x7f898ea02D2f8113b92035D78f9073040a545696	150.98 ETH	4	1	
0x74210C2a4B7E3bcbBeB288c6B1B25E50c2E86d27	100.00 ETH	0	2	
0xBe3EB6a8d70F9Ab81775bB132B9FE168Dd96F2CF	100.00 ETH	0	3	



PNP TOKEN ICO SALE

Introducing "PNP Token" (PNP)! Token price is 0.001 Ether. You currently have 2000 PNP.

1000 | Buy Tokens

2000 / 750000 tokens sold

Your Account: 0x7f898ea02d2f8113b92035d78f9073040a545696

```

    ▲ ATTENTION: In an effort to improve user privacy, inpage.js:1
    MetaMask will stop exposing user accounts to dapps by default
    beginning November 2nd, 2018. Dapps should call
    provider.enable() in order to view and use accounts. Please
    see https://bit.ly/200HXvF for complete information and up-to-
    date example code.
    App initialized... app.js:11
    Pnp Token Sale Address: app.js:33
    0xb4aae47d878df3e49611fb7ef8842fc0406e0c1f
    Pnp Token Address: app.js:40
    0x80029252fcfd540dbaf5ed5e3862fa4ac34e0ed2
    Tokens bought... app.js:122
    event triggered app.js:56
    {logIndex: 1, transactionIndex: 0, transactionHash: "0xec719
    1e4ecc0a2bc66d36346669158eccf5bf7b34443681eb280504dd8adadb0"
    , blockHash: "0xad6d682268ffe2f535d3e5c473602a8f458ccf47f28e
    2f4052c70667b95b480c", blockNumber: 30, ...}
    >
    Tokens bought... app.js:122
    event triggered app.js:56
    {logIndex: 1, transactionIndex: 0, transactionHash: "0xb1e93
    c4f5871cd9c136ed5118b93c47ffcb81b6b6e74fb2130f02c56a22bbb3d"
    , blockHash: "0xbaa03d2391382803660ac0d4235cec55d1217eb3c8e9
    10513443f9f4ff958679", blockNumber: 31, ...}
    >
  
```

Confirmed transaction
Transaction 5 confirmed! View on EtherScan

```

⚠ ▶ ATTENTION: In an effort to improve user privacy, inpage.js:1
MetaMask will stop exposing user accounts to dapps by default
beginning November 2nd, 2018. Dapps should call
provider.enable() in order to view and use accounts. Please
see https://bit.ly/2Q0HXvF for complete information and up-to-
date example code.

App initialized...                                         app.js:11
Pnp Token Sale Address:                                app.js:33
0xb4ae47d878df3e49611fbf7ef8842fc0406e0c1f

Pnp Token Address:                                     app.js:40
0x80029252fcfd540dbaf5ed5e3862fa4ac34e0ed2

Tokens bought...                                       app.js:122
event triggered                                         app.js:56
{logIndex: 1, transactionIndex: 0, transactionHash: "0xec719
1e4ecc0a2bc66d36346669158eccf5bf7b34443681eb280504dd8adadb0"
, blockHash: "0xad6d682268ffe2f535d3e5c473602a8f458ccf47f28e
2f4052c70667b95b480c", blockNumber: 30, ...}

Tokens bought...                                       app.js:122
event triggered                                         app.js:56
{logIndex: 1, transactionIndex: 0, transactionHash: "0xb1e93
c4f5871cd9c136ed5118b93c47ffc8b1b6b6e74fb2130f02c56a22bbb3d"
, blockHash: "0xbaa03d2391382803660ac0d4235cec55d1217eb3c8e9
10513443f9f4ff958679", blockNumber: 31, ...} ⓘ
address: "0xb4ae47d878df3e49611fbf7ef8842fc0406e0c1f"

▼ args:
▶ _amount: r {s: 1, e: 3, c: Array(1)}
  _buyer: "0x7f898ea02d2f8113b92035d78f9073040a545696"
▶ __proto__: Object
blockHash: "0xbaa03d2391382803660ac0d4235cec55d1217eb3c8e..."
blockNumber: 31
event: "Sell"
logIndex: 1
transactionHash: "0xb1e93c4f5871cd9c136ed5118b93c47ffc8b1...
transactionIndex: 0
type: "mined"
▶ __proto__: Object

```

Ganache					
ACCOUNTS	BLOCKS	TRANSACTIONS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES	
CURRENT BLOCK 32	GAS PRICE 2000000000	GAS LIMIT 6721975	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING
MNEMONIC <small>?</small>					HD PATH m/44'/60'/0'/0/account_index
armed artist bar chimney journey disease mass cruel sense whisper mix latin					
ADDRESS 0x001e54cE86bAB90f73E8667E63fb02BDA9A83916		BALANCE 48.22 ETH		TX COUNT 25	INDEX 0
ADDRESS 0x7f898ea02D2f8113b92035D78f9073040a545696		BALANCE 147.98 ETH		TX COUNT 7	INDEX 1
ADDRESS 0x74210C2a4B7E3bcbBeB288c6B1B25E50c2E86d27		BALANCE 100.00 ETH		TX COUNT 0	INDEX 2
ADDRESS 0xBe3EB6a8d70F9Ab81775bB132B9FE168Dd96F2CF		BALANCE 100.00 ETH		TX COUNT 0	INDEX 3

Ganache					
ACCOUNTS	BLOCKS	TRANSACTIONS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES	
CURRENT BLOCK 32	GAS PRICE 2000000000	GAS LIMIT 6721975	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING
BLOCK 32	MINED ON 2018-10-29 13:27:07			GAS USED 46407	1 TRANSACTION
BLOCK 31	MINED ON 2018-10-29 13:25:52			GAS USED 46407	1 TRANSACTION
BLOCK 30	MINED ON 2018-10-29 13:25:16			GAS USED 76407	1 TRANSACTION
BLOCK 29	MINED ON 2018-10-29 13:21:28			GAS USED 21000	1 TRANSACTION
BLOCK 28	MINED ON 2018-10-29 13:15:58			GAS USED 51258	1 TRANSACTION
BLOCK 27	MINED ON 2018-10-29 13:09:14			GAS USED 21000	1 TRANSACTION
BLOCK 26	MINED ON 2018-10-29 13:07:16			GAS USED 27008	1 TRANSACTION
BLOCK	MINED ON			GAS USED	

The screenshot shows a web-based Ethereum interface with the following details:

- Current Block:** 32
- GAS PRICE:** 2000000000
- GAS LIMIT:** 6721975
- NETWORK ID:** 5777
- RPC SERVER:** HTTP://127.0.0.1:7545
- MINING STATUS:** AUTOMINING

TX HASH: **0xffff4f0387fe7edfddabe66ac76feeb58c456dd9ea2bb7ea58ecbbfef749950c8c**

FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0xf898ea02D2f8113b92035D78f9073040a545696	0xb4ae47D878df3e49611fBf7ef8842FC0406e0C1F	46407	10000000000000000000000000000000

TX HASH: **0xb1e93c4f5871cd9c136ed5118b93c47ffc8b1b6b6e74fb2130f02c56a22bbb3d**

FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0xf898ea02D2f8113b92035D78f9073040a545696	0xb4ae47D878df3e49611fBf7ef8842FC0406e0C1F	46407	10000000000000000000000000000000

TX HASH: **0xec7191e4ecc0a2bc66d36346669158eccf5bf7b34443681eb280504dd8adadb0**

FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0xf898ea02D2f8113b92035D78f9073040a545696	0xb4ae47D878df3e49611fBf7ef8842FC0406e0C1F	76407	10000000000000000000000000000000

TX HASH: **0x185f77215002b44bbdbcfacdf2250205ab17ed21767c608974e215f22eb6311e**

FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
--------------	---------------------	----------	-------

The screenshot shows the Ganache interface with the following details:

- CURRENT BLOCK:** 32
- GAS PRICE:** 2000000000
- GAS LIMIT:** 6721975
- NETWORK ID:** 5777
- RPC SERVER:** HTTP://127.0.0.1:7545
- MINING STATUS:** AUTOMINING

LOGS:

```
[12:36:55 PM] Starting server (version 1.2.2) with initial configuration:
{"hostname":"127.0.0.1","port":7545,"network_id":5777,"default_balance_ether":100,"total_accounts":10,"unlocked_accounts":[],"locked":false,"vmErrorsOnRPCResponse":true,"verbose":false}
[12:36:55 PM] Ganache started successfully!
[12:36:55 PM] Waiting for requests ...
[12:42:59 PM] net_version
[12:42:59 PM] eth_accounts
[12:42:59 PM] eth_accounts
[12:42:59 PM] eth_accounts
[12:42:59 PM] eth_sendTransaction
[12:43:00 PM] Transaction: 0x33824209d293a0833436f36febe446312f13db7ed5e8042d2ac40b4b331373df
[12:43:00 PM] Contract created: 0xc0b863e1e3f6550c130b0ef145bf6194066b59ae
[12:43:00 PM] Gas usage: 277462
[12:43:00 PM] Block Number: 1
[12:43:00 PM] Block Time: Mon Oct 29 2018 12:42:59 GMT+0530 (India Standard Time)
[12:43:00 PM] eth_newBlockFilter
[12:43:00 PM] eth_getFilterChanges
[12:43:00 PM] eth_getTransactionReceipt
[12:43:00 PM] eth_getCode
[12:43:00 PM] eth_uninstallFilter
[12:43:00 PM] eth_sendTransaction
[12:43:00 PM] Transaction: 0x20d18b5fe5d0e5faa7ab118ab942bf3f477a6fe92b596c1a027cd1489c8a1870
[12:43:00 PM] Gas usage: 42008
[12:43:00 PM] Block Number: 2
```

Session 14: Deploying the Smart Contracts with Geth (Not Fully Completed)

1. Install Geth from <https://geth.ethereum.org/downloads/>

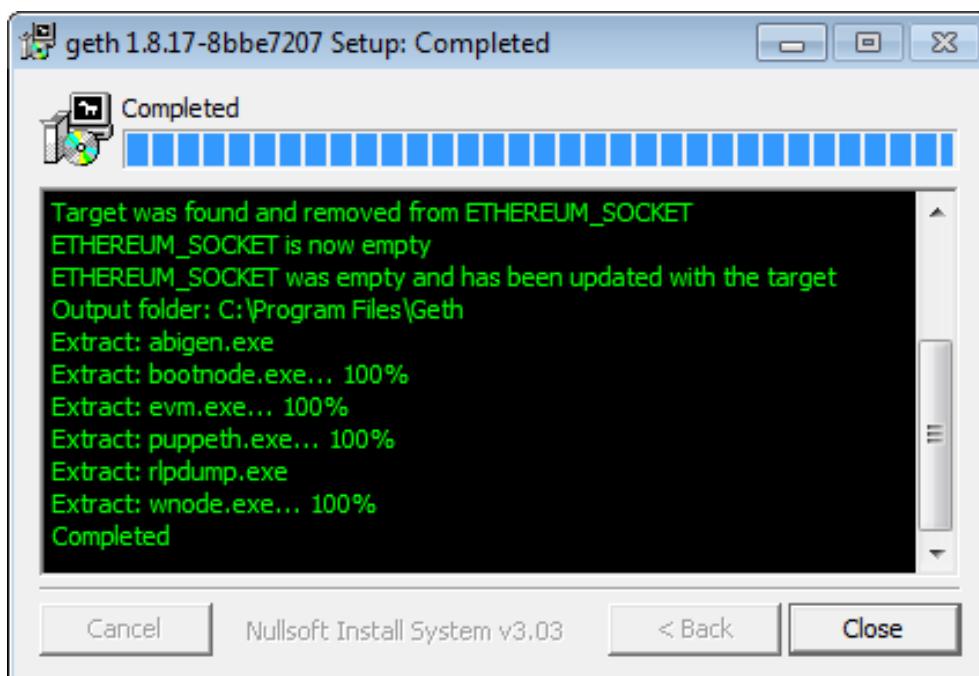
The screenshot shows a web browser displaying the Ethereum Geth download page at <https://geth.ethereum.org/downloads/>. The page title is "Download Geth – Shoutingstone (v1.8.17) – Release Notes". Below the title, there is a paragraph of text: "You can download the latest 64-bit stable release of Geth for our primary platforms below. Packages for all supported platforms, as well as develop builds, can be found further down the page. If you're looking to install Geth and/or associated tools via your favorite package manager, please check our [installation guide](#)". Below the text are four download links: "Geth 1.8.17 for Linux" (with a Linux icon), "Geth 1.8.17 for macOS" (with a Mac icon), "Geth 1.8.17 for Windows" (with a Windows icon), and "Geth 1.8.17 sources" (with a source code icon). The browser's address bar shows the URL, and the top navigation bar has various bookmarked items.

Specific Versions

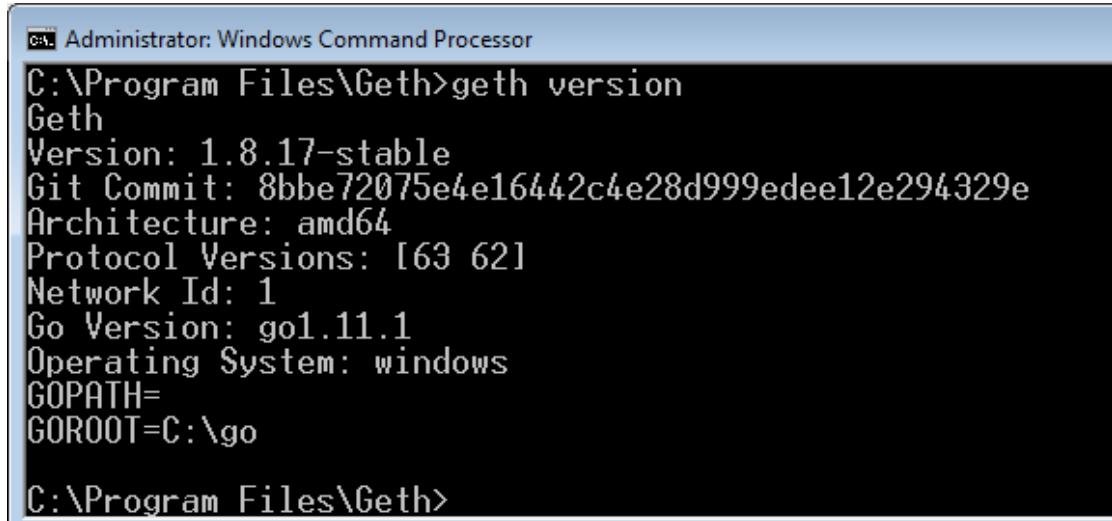
If you're looking for a specific release, operating system or architecture, below you will find:

- All stable and develop builds of Geth and tools
- Archives for non-primary processor architectures
- Android library archives and iOS XCode frameworks

Please select your desired platform from the lists below and download your bundle of choice. Please be aware that the [MD5](#) checksums are provided by our binary hosting platform (Azure Blobstore) to help check for download errors. For security guarantees please verify any downloads via the attached PGP signature files (see [OpenPGP Signatures](#) for details).



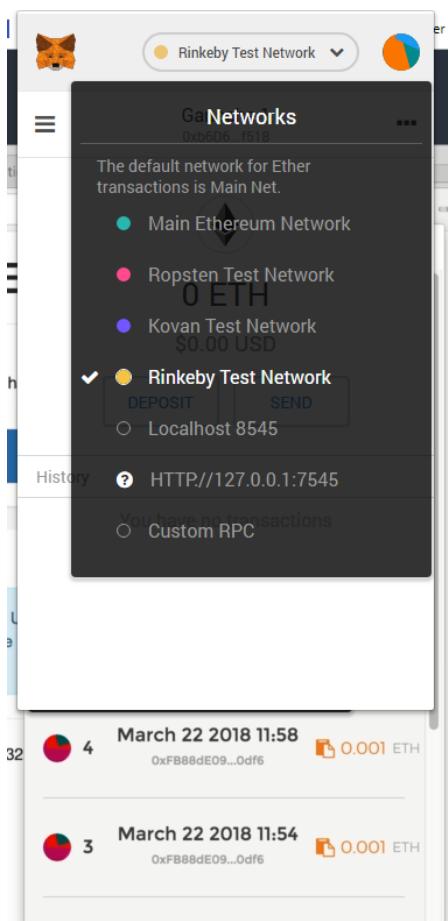
2. Open console & go to geth path



```
Administrator: Windows Command Processor
C:\Program Files\Geth>geth version
Geth
Version: 1.8.17-stable
Git Commit: 8bbe72075e4e16442c4e28d999edee12e294329e
Architecture: amd64
Protocol Versions: [63 62]
Network Id: 1
Go Version: go1.11.1
Operating System: windows
GOPATH=
GOROOT=C:\go

C:\Program Files\Geth>
```

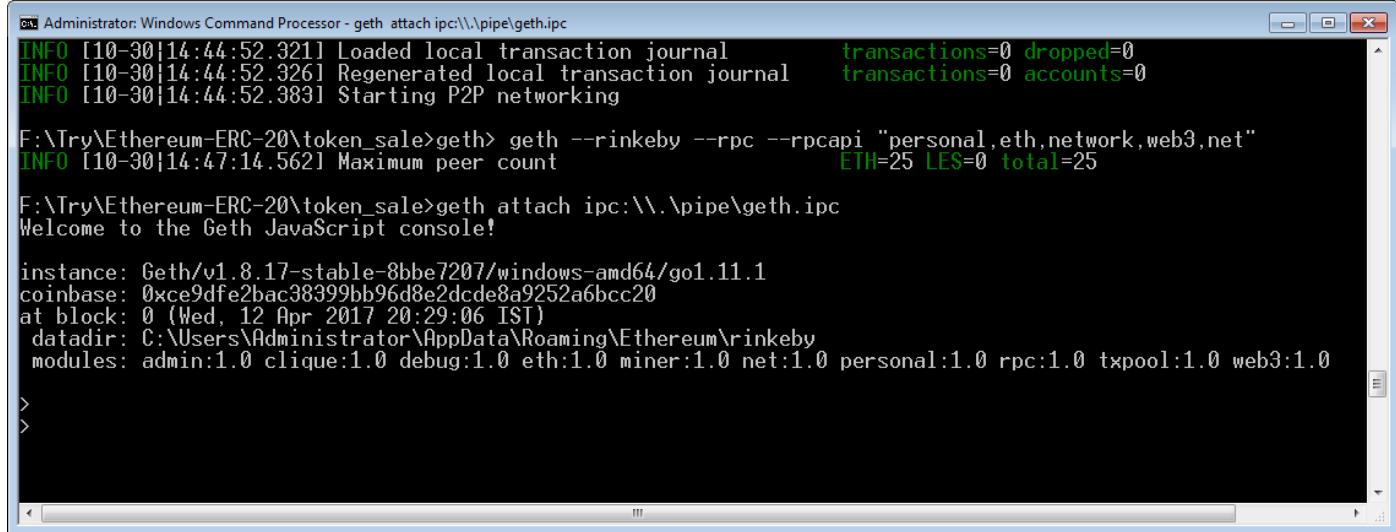
3. Go to Browser & select Rinkeby Test Network



4. Go to console (geth path) and write following comma

```
geth --rinkeby --rpc --rpcapi="personal,eth,network,web3,net" or
```

geth --rinkeby --rpc --rpccapi="personal,eth,network,web3,net" --ipcpath "PATH in Hint"
 (Hint: **OS X** ~/Library/Ethereum/ **Linux** ~/.ethereum/ **Windows**
 ~/AppData/Roaming/Ethereum)



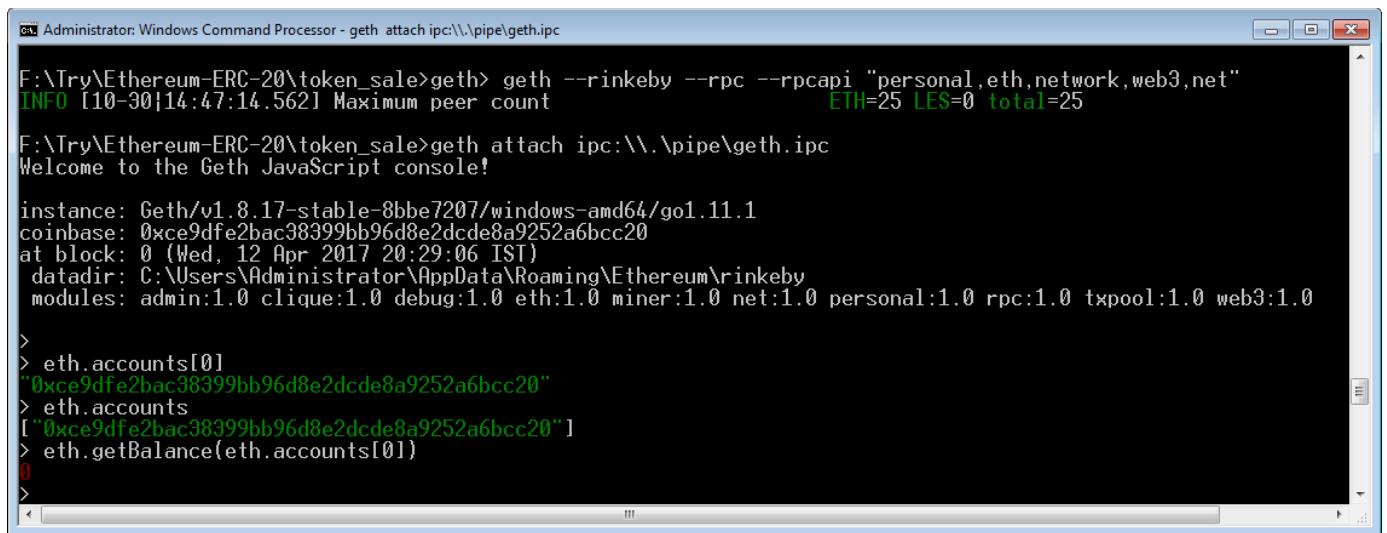
```
Administrator: Windows Command Processor - geth attach ipc:\\\\pipe\\geth.ipc
INFO [10-30|14:44:52.321] Loaded local transaction journal           transactions=0 dropped=0
INFO [10-30|14:44:52.326] Regenerated local transaction journal      transactions=0 accounts=0
INFO [10-30|14:44:52.383] Starting P2P networking

F:\Try\Ethereum-ERC-20\token_sale>geth --rinkeby --rpc --rpccapi "personal,eth,network,web3,net"
INFO [10-30|14:47:14.562] Maximum peer count                      ETH=25 LES=0 total=25

F:\Try\Ethereum-ERC-20\token_sale>geth attach ipc:\\\\pipe\\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.8.17-stable-8bbe7207/windows-amd64/go1.11.1
coinbase: 0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20
at block: 0 (Wed, 12 Apr 2017 20:29:06 IST)
datadir: C:\Users\Administrator\AppData\Roaming\Ethereum\rinkeby
modules: admin:1.0 clique:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

>
```



```
Administrator: Windows Command Processor - geth attach ipc:\\\\pipe\\geth.ipc
F:\Try\Ethereum-ERC-20\token_sale>geth --rinkeby --rpc --rpccapi "personal,eth,network,web3,net"
INFO [10-30|14:47:14.562] Maximum peer count                      ETH=25 LES=0 total=25

F:\Try\Ethereum-ERC-20\token_sale>geth attach ipc:\\\\pipe\\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.8.17-stable-8bbe7207/windows-amd64/go1.11.1
coinbase: 0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20
at block: 0 (Wed, 12 Apr 2017 20:29:06 IST)
datadir: C:\Users\Administrator\AppData\Roaming\Ethereum\rinkeby
modules: admin:1.0 clique:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

>
> eth.accounts[0]
"0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20"
> eth.accounts
["0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20"]
> eth.getBalance(eth.accounts[0])
0
>
```

The screenshot shows the Etherscan interface for the Rinkeby testnet. The top navigation bar includes links for Google, Inbox, Release 6_deploy, PNP Token ICO S, go ethereum - Fu, Code Your Own C, TESTNET Rinkeby, and Other bookmarks. The main header features the Etherscan logo and the text "RINKEBY (CLIQUE) TESTNET". Below the header, there's an advertisement for "UPGRADE PAYCHECK" with a "START LEARNING" button. The main content area is divided into two sections: "Blocks" on the left and "Transactions" on the right.

Blocks:

- Block 3228859: Mined By 0xd6ae8250b8348c..., 20 txns in 15 secs, Block Reward 0 Ether, > 36 secs ago
- Block 3228858: Mined By 0xb279182d99e657..., 18 txns in 15 secs, Block Reward 0 Ether, > 51 secs ago
- Block 3228857: Mined By 0x7ffc57839b00206..., 29 txns in 15 secs, Block Reward 0 Ether, > 1 min ago
- Block 3228856: Mined By 0x635f83421bf059..., 38 txns in 15 secs, Block Reward 0 Ether, > 1 min ago
- Block 3228855: Mined By 0x42eb768f2244c88..., 21 txns in 15 secs, Block Reward 0 Ether, > 1 min ago
- Block 3228854: Mined By 0xfc18cbc391de84d..., 16 txns in 15 secs, Block Reward 0 Ether, > 1 min ago

Transactions:

- TX# 0X1FFC2375A854B99BB7E9EC9... From 0x195a07037e97cd... To 0xe93ab27a89a415... Amount 0 Ether, > 36 secs ago
- TX# 0XF2466AA8E9A5FED269D4C98... From 0x8a37b79e54d69e... To 0x54a298ee9fccbf0... Amount 0 Ether, > 36 secs ago
- TX# 0X81F93416470BADBF090288B... From 0xea654bd738207c... To 0x3ce0d0805d50f97... Amount 0 Ether, > 36 secs ago
- TX# 0XB6C81DBA8D75242D0E5AE8... From 0xbe1085bc3e0812f... To 0x40af244c94e679a... Amount 0 Ether, > 36 secs ago
- TX# 0X91A03EB1BC3DFF0563ECD8F... From 0x903f5a9ecadb63b... To [NewContract] Amount 0 Ether, > 36 secs ago
- TX# 0XF5FC9FD04181B5E7FEE6494... From 0x308f27c8595b2ee... To 0xb944c7a25e813a... Amount 0.1 Ether, > 36 secs ago

A message at the bottom states: "This Website uses cookies to improve your experience. And has an updated Privacy Policy." A "Got It" button is also present.

```
C:\Program Files\Geth>geth --rinkeby account new
INFO [10-27|09:34:34.595] Maximum peer count          ETH=25 LES=0 total=25
Your new account is locked with a password. Please give a password. Do not forget this password.
Passphrase:
Repeat passphrase:
Address: {1c1e1ae6887c238d37fb01aa186d492e85a809c4}

C:\Program Files\Geth>
```

```
F:\Try\Ethereum-ERC-20\token_sale>geth --rinkeby account new
INFO [10-29|13:46:44.731] Maximum peer count          ETH=25 LES=0 total=25
Your new account is locked with a password. Please give a password. Do not forget this password.
Passphrase:
Repeat passphrase:
Address: {ce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20}
```

F:\Try\Ethereum-ERC-20\token_sale>

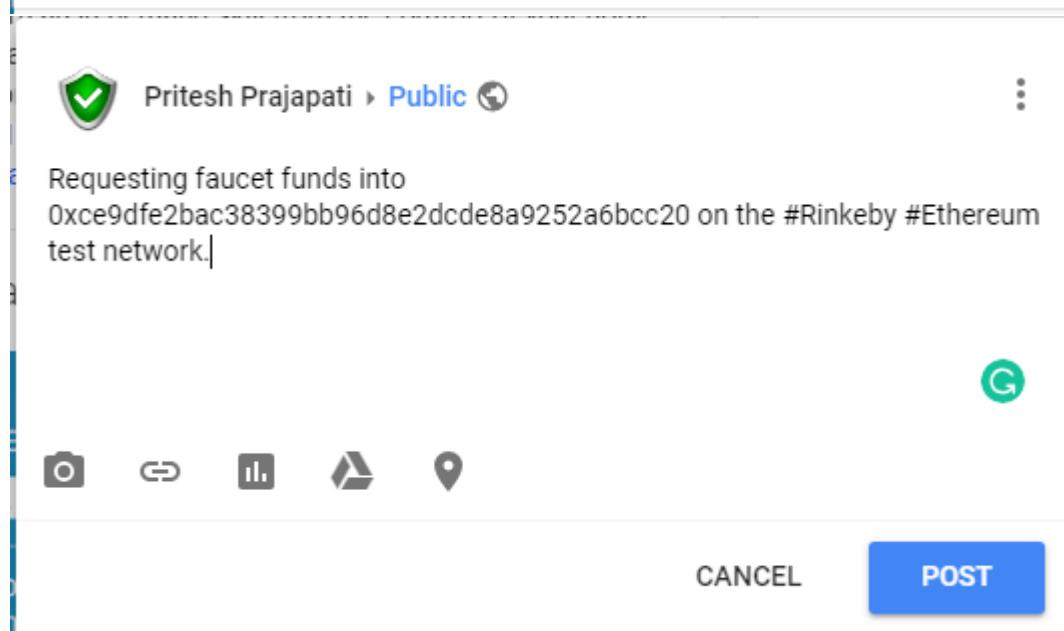
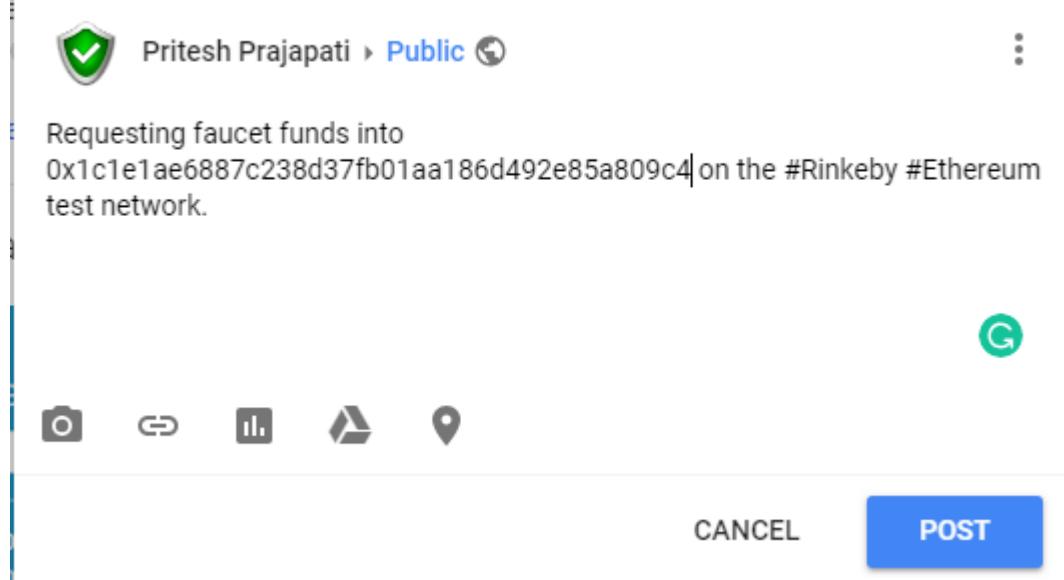
Password: pritesh

Request Ether from <https://faucet.rinkeby.io/>

For Address: 1c1e1ae6887c238d37fb01aa186d492e85a809c4

For Address: ce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20

Post in Google Plus Account



Faucet

Rinkeby Authenticated Faucet

<https://plus.google.com/+PriteshPrajapati007/posts/8VgRBKRAE7i> Give me Ether ▾

0x1c1e1ae6887c238d37fb01aa186d492e85a809c4 funded

8 peers 3232364 blocks 9.046256971665328e+56 Ethers 212136 funded

Search For Address: 1c1e1ae6887c238d37fb01aa186d492e85a809c4

Search For Address: ce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20

The screenshot shows the Etherscan.io interface for the RINKEBY (CLIQUE) TESTNET. The address 0x1c1e1ae6887c238d37fb01aa186d492e85a809c4 is being searched. The address overview shows a balance of 18.75 Ether and one transaction. The transaction details are as follows:

TxHash	Block	Age	From	To	Value	[TxFee]	
0x0f805fc74fb2b66d...	3232364	15 mins ago	0x31b98d14007bde...	IN	0x1c1e1ae6887c23...	18.75 Ether	0.000021



Address 0x1C1E1Ae6887C238d37fb01AA186d492e85A809C4



Overview



Balance: 18.75 Ether

Transactions: 1 txn

Overview

Transaction Information

[This is a Rinkeby Testnet Transaction Only]

TxHash: 0x0f805fc74fb2b66d2214a45ce2a307033cfcb394d0c611767d1fa4dff2ff40ef

TxReceipt Status: Success

Block Height: 3232364 (74 Block Confirmations)

TimeStamp: 18 mins ago (Oct-27-2018 04:18:21 AM +UTC)

From: 0x31b98d14007bdee637298086988a0bbd31184523

To: 0x1c1e1ae6887c238d37fb01aa186d492e85a809c4

Value: 18.75 Ether (\$0.00)

Gas Limit: 21000

Gas Used By Transaction: 21000

Gas Price: 0.000000001 Ether (1 Gwei)

Actual Tx Cost/Fee: 0.000021 Ether (\$0.000000)

Nonce & {Position}: 212135 | {9}

Input Data:

0x

Overview	
Block Information	
Height:	3232364
TimeStamp:	19 mins ago (Oct-27-2018 04:18:21 AM +UTC)
Transactions:	13 transactions and 0 contract Internal Transaction in this Block
Hash:	0x944373e19bf9af402d66d1f4851d5ae819803aa16d59b58282fd6c883268351d
Parent Hash:	0xa290651e44f6c5f716125d9cc3d02b709df61b08daa3b96ac964a279752ac51a
Sha3Uncles:	0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347
Mined By:	0x7ffc57839b00206d1ad20c69a1981b489f772031 in 15 secs
Difficulty:	2
Total Difficulty:	5,982,818
Size:	5670 bytes
Gas Used:	5,371,630 (69.09%)
Gas Limit:	7,774,976
Nonce:	0x0000000000000000
Block Reward:	0 Ether
Uncles Reward:	0
Extra Data:	<pre>010811/geth/go1.11.1/linux (Hex:0xd883010811846765746888676f312e31312e31856c696e757800000000000001d24042d124e6ae387cd3004645159b7e3d4e8 c06d4241e17f9b2e10f868460812a2cd032664ddbbc1827feaaa9839b4e67d5b71795adbb46c7add6eec14125c00)</pre>

For Account 2

Etherscan RINKEBY (CLIQUE) TESTNET Search by Address / Txhash / Block / Token / Ens GO Language

Address 0xcE9DfE2BaC38399bb96d8e2dcDE8A9252a6bCc20

HOME BLOCKCHAIN TOKEN MISC

Home / Accounts / Address

Overview	More Options					
Balance: 18.75 Ether						
Transactions: 1 txn						
Transactions						
1 Latest 1 txn						
TxHash	Block	Age	From	To	Value	[TxFee]
0x686f0d44af7e54c...	3244886	2 mins ago	0x31b98d14007bde...	IN	0xce9dfe2bac38399...	18.75 Ether 0.000021

Ethereum-ERC-20

Mr. Priteshkumar Prajapati

Overview	
Transaction Information	Home / Transactions / Tx Info
[This is a Rinkeby Testnet Transaction Only]	
TxHash:	
TxReceipt Status:	Success
Block Height:	3244886 (14 Block Confirmations)
TimeStamp:	4 mins ago (Oct-29-2018 08:28:56 AM +UTC)
From:	0x31b98d14007bdee637298086988a0bbd31184523
To:	0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20
Value:	18.75 Ether (\$0.00)
Gas Limit:	21000
Gas Used By Transaction:	21000
Gas Price:	0.000000001 Ether (1 Gwei)
Actual Tx Cost/Fee:	0.000021 Ether (\$0.000000)
Nonce & {Position}:	213019 {13}
Input Data:	0x

Address

0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20

Go!

rinkeby (infura)

Address: 0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20
 Balance: 18.75 Ether
 USD Value: \$ 3813.19 (@203.37/Eth)
 EUR Value: € 3330.94 (@177.65/Eth)
 BTC Value: 0.59 Btc (@0.03156/Eth)

rinkeby (infura)

Address: 0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20
 Balance: 18.75 Ether
 USD Value: \$ 3813.19 (@203.37/Eth)
 EUR Value: € 3330.94 (@177.65/Eth)
 BTC Value: 0.59 Btc (@0.03156/Eth)

Price data Source: [CryptoCompare](#)

Address 0xE9DfE2BaC38399bb96d8e2dcDE8A9252a6bCc20 

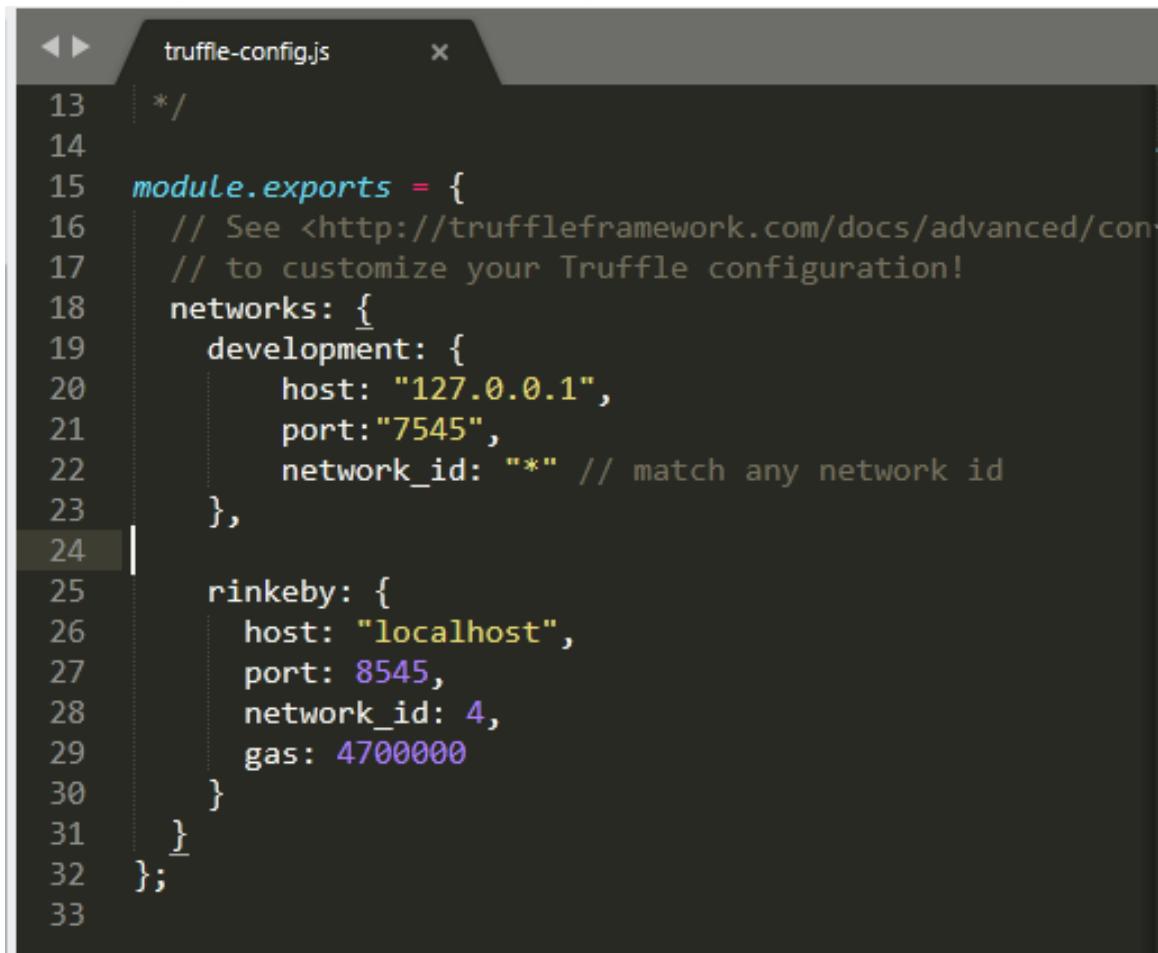
[Home](#) / [Accounts](#) / [Address](#)

Overview		More Options 
Balance:	21.75 Ether	
Transactions:	2 txns	

Transactions

Latest 2 txns							
TxHash	Block	Age	From	To	Value	[TxFee]	
0x97827a7d186036...	3250891	31 mins ago	0x31b98d14007bde...	IN	0xce9dfe2bac38399...	3 Ether	0.000021
0x686f0d44af7e54c...	3244886	1 day 1 hr ago	0x31b98d14007bde...	IN	0xce9dfe2bac38399...	18.75 Ether	0.000021

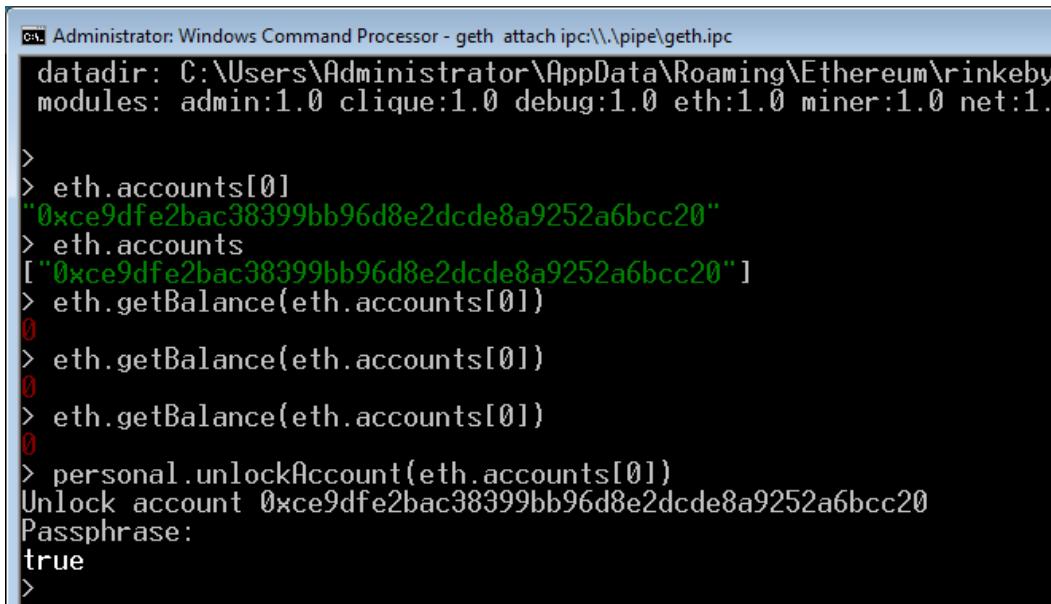
2. Now modify the code of “truffle-config.js”



```
truffle-config.js
```

```
13  */
14
15 module.exports = {
16   // See <http://truffleframework.com/docs/advanced/con-
17   // to customize your Truffle configuration!
18   networks: {
19     development: {
20       host: "127.0.0.1",
21       port: "7545",
22       network_id: "*" // match any network id
23     },
24
25     rinkeby: {
26       host: "localhost",
27       port: 8545,
28       network_id: 4,
29       gas: 4700000
30     }
31   }
32 };
33
```

3. Goto console & Unlock the account (**Hint:** geth attach ipc:\\.\pipe\geth.ipc)

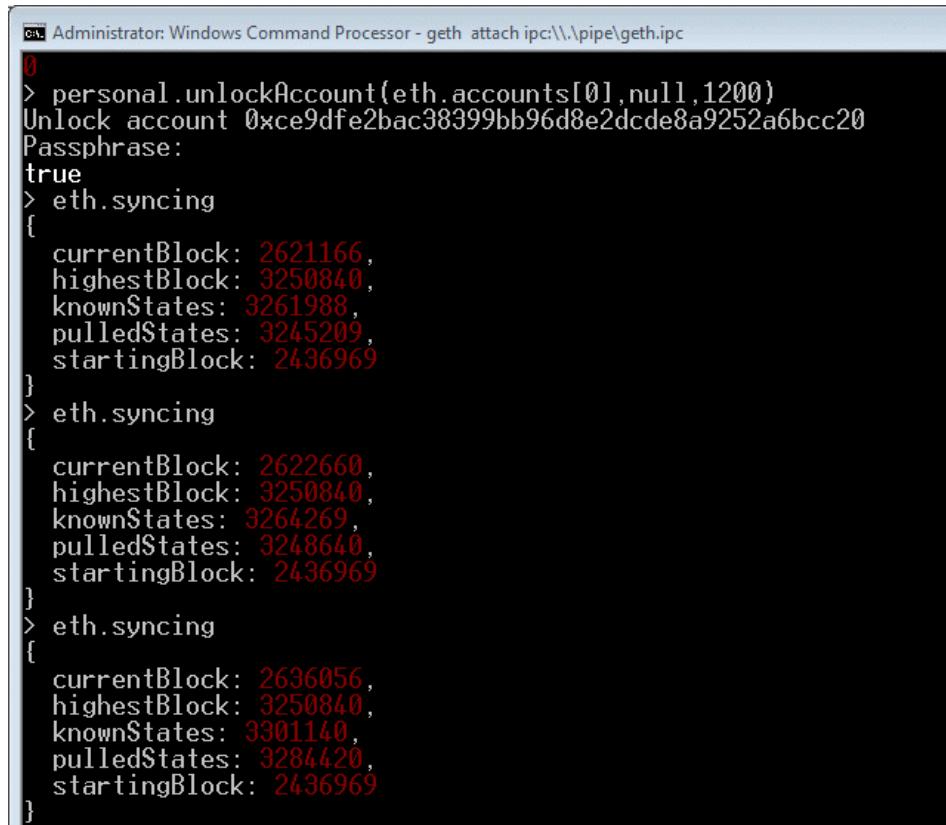


```
Administrator: Windows Command Processor - geth attach ipc:\\.\pipe\geth.ipc
datadir: C:\Users\Administrator\AppData\Roaming\Ethereum\rinkeby
modules: admin:1.0 clique:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.

>
> eth.accounts[0]
"0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20"
> eth.accounts
["0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20"]
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[0])
0
> personal.unlockAccount(eth.accounts[0])
Unlock account 0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20
Passphrase:
true
>
```

```
> personal.unlockAccount(eth.accounts[0],null,1200)
Unlock account 0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20
Passphrase:
true
>
```

4. Wait till currentBlock=highestBlock (Not Fully Completed but you can try !!!)



```
Administrator: Windows Command Processor - geth attach ipc:\\\\.\pipe\\geth.ipc
0
> personal.unlockAccount(eth.accounts[0],null,1200)
Unlock account 0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20
Passphrase:
true
> eth.syncing
{
  currentBlock: 2621166,
  highestBlock: 3250840,
  knownStates: 3261988,
  pulledStates: 3245209,
  startingBlock: 2436969
}
> eth.syncing
{
  currentBlock: 2622660,
  highestBlock: 3250840,
  knownStates: 3264269,
  pulledStates: 3248640,
  startingBlock: 2436969
}
> eth.syncing
{
  currentBlock: 2636056,
  highestBlock: 3250840,
  knownStates: 3301140,
  pulledStates: 3284420,
  startingBlock: 2436969
}
```

```
Administrator: Windows Command Processor - geth attach ipc:\\.\pipe\geth.ipc
> eth.syncing
{
  currentBlock: 3251066,
  highestBlock: 3251169,
  knownStates: 3980724,
  pulledStates: 3979587,
  startingBlock: 2436969
}
> eth.syncing
{
  currentBlock: 3251066,
  highestBlock: 3251169,
  knownStates: 3980724,
  pulledStates: 3979587,
  startingBlock: 2436969
}
> eth.syncing
{
  currentBlock: 3251066,
  highestBlock: 3251169,
  knownStates: 3980724,
  pulledStates: 3979587,
  startingBlock: 2436969
}
>
```

```
Administrator: Windows Command Processor - geth attach ipc:\\.\pipe\geth.ipc
F:\Try\Ethereum-ERC-20\token_sale>geth> geth --rinkeby --rpc --rpccapi "personal,eth,network,web3,net"
INFO [10-30|14:47:14.562] Maximum peer count                                     ETH=25 LES=0 total=25
F:\Try\Ethereum-ERC-20\token_sale>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.8.17-stable-8bbe7207/windows-amd64/go1.11.1
coinbase: 0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20
at block: 0 (Wed, 12 Apr 2017 20:29:06 IST)
datadir: C:\Users\Administrator\AppData\Roaming\Ethereum\rinkeby
modules: admin:1.0 clique:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0
>
> eth.accounts[0]
"0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20"
> eth.accounts
["0xce9dfe2bac38399bb96d8e2dcde8a9252a6bcc20"]
> eth.getBalance(eth.accounts[0])
0
>
```

geth attach ipc:\\.\pipe\geth.ipc

5. Truffle migrate --reset --compile-all --network rinkeby
 6. geth attach ipc:\\.\pipe\geth.ipc
 7. personal.unlockAccount(eth.accounts[0],null,1200)
 8. Truffle migrate --reset --compile-all --network rinkeby
 9. Go to Build>contracts>PnpToken.json
 10. Go to <https://rinkeby.etherscan.io> and verify by Block/Address
 11. Go to Build>contracts>PnpTokenSale.json
 12. Go to <https://rinkeby.etherscan.io> and verify by Block/Address
 13. Go to Console geth attach ipc:\\.\pipe\geth.ipc
- ```
var admin=eth.accounts[0]
```

```
var tokensAvailable = 75000
tokensAvailable
var tokenSaleAddress='address' >build>PnpTokenSale.json networks 4 address
tokenSaleAddress
```