Truffle:

a) **Is a framework that helps developers with Testing, Deployment and Management of Smart Contracts and Distributed Applications**
b) Is a library that helps developers to connect to Ethereum nodes, because it abstracts the JSON-RPC interface
c) Is a framework for Java, similar to Web3.js for JavaScript. It's a great way to develop distributed java enterprise applications.

Unit-Testing on a local chain is important, because it helps you to

a) **Run tests quickly and especially for free, compared to continuous deployment on the Main-Network. This way you save a lot of fees, time and costs.**
b) Run tests is an environment where logging is activated. On the Main-Net you have no access to transaction logs and this is ultimately the information you need to debug your contracts
c) Avoid regression bugs with contracts that are updated constantly on the main-net. Once you update a contract on the main-net, the address stays the same, but the code changes and this can have disastrous side-effects.

With truffle it's easy to write clean-room unit-tests

a) **For Solidity and JavaScript**
b) For Java, JavaScript, and C++
c) For JavaScript using Web3.js
d) For any language, as long as it adheres to the open Testing-Interface from Truffle.

With the truffle config file you can manage

a) **Different Networks to deploy your contracts to. This way you can easily deploy to a local blockchain, the main-net or the Ropsten/Rinkeby Test-Net with only one parameter.**
b) The amount of gas your contract deployment and transactions against your contract will need. This way you can essentially lower the gas costs over traditional web3.js dApps.
c) You can manage your secret API keys to the Ethereum Network. This way you can get access to several different Ethereum nodes at the same time without the need to switch your keyfiles.