

# JJSWebEx Manual

Acos-server is a Node.js application currently used as a web service to fetch the content from the database and display it in the web browser.

Requirements:

**IDE:** IntelliJ IDEA (<https://www.jetbrains.com/idea/download>)

**PHP Server:** Xampp ([link](#)) or PHP ([link](#))

**Databases:** aggregate, webex21    **Database Server:** MySQL Server

## Installing

1. Download the git repository from the following link and import it into the IDE:  
<https://github.com/PAWSLabUniversityOfPittsburgh/jswebex.git>
2. Copy paste the file **database\_fetch.php** from jswebex/public folder to the htdocs folder of either xampp or the PHP directory
3. Change the username and the password of the database from the **database\_fetch.php** file to connect to your localhost database
4. Then run the server by running the **jswebex/bin/www** file either from the IDE or from the command prompt/terminal window by going into the bin directory and typing the command '**node www**'
5. Overall: Make sure you have the server running (as per step 4), the apache server running for your PHP file to work and your MySQL server running for your database to function.
6. Then you may access any of the examples from your browser by using the following link for example:

<http://localhost:3000/pitt/annotated/annotated-demo/foo?act=pyt3.2&usr=me&grp=001&sid=001>

The same example can be accessed on the server by the following link for example:

<http://adapt2.sis.pitt.edu/pitt/annotated/annotated-demo/foo?act=pyt3.2&usr=me&grp=001&sid=001>

Here the examples are identified by the act parameter of the URL which denotes the rdfID of the example.

Link Parameters: Possible link (URL) parameters:

- sub – denotes line number – is always '0' when the example is opened
- act - denotes rdfid of the example
- sid - session id
- usr = user id
- res - is always '1'

- grp = group id
- svc - is always 'jswebex'
- app id = is always '3'

## File Structure

- acos-server/bin/www – Run the file to start the server process
- acos-server/node\_modules/acos-annotated/static/annotated.js – Level 2 parser
- acos-server/node\_modules/acos-annotated/index.js – The page scripts for the web interface (individual example layout)
- acos-server/node\_modules/acos-annotated-demo/static/example.js – Retrieves the entire json object from the database using the php file call and also does the Level 1 parsing.
- acos-server/node\_modules/acos-annotated-demo/index.js – Homepage (back-end logic)
- acos-server/node\_modules/acos-pitt – handles the pitt server parametric information
- acos-server/public/test.php – File that retrieves the entire object from the webex database

## Homepage Access:

<http://localhost:3000/>

Python Examples

Variables

1. Hello World  
2. Different Ways to Print  
3. Variable Assignment  
4. Multiple Variable Assignment  
5. Variable type: Numbers  
6. Variable type: Strings  
7. printing.py  
8. Simple Calculations  
9. Arithmetic Operations  
10. Assignment Operators

Simple Input

Loops

Strings [Close](#)

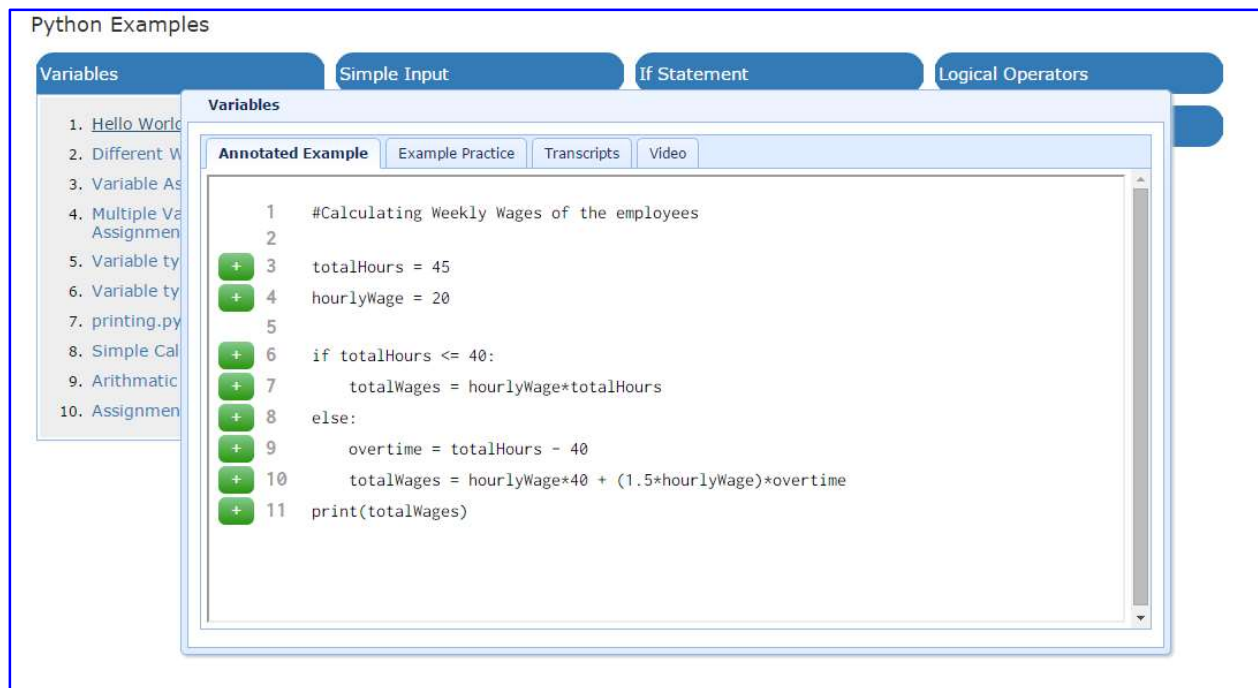
If Statement

Functions

Logical Operators

Lists

## Example Access:



## Ports:

The acos-server is initiated at port 3000. The default HTTP listen is on port 8080. The SQL server should be on port 3306.

## Node Modules Details:

- express - Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications
- nunjucks – templating module for the front-end interface (views)
- body-parser – This is a node.js middleware for handling JSON, Raw, Text and URL encoded form data
- cookie-parser - Parse Cookie header and populate req.cookies with an object keyed by the cookie names
- colors – used to display colored messages in the console
- morgan – console logger (of events)

## Link Meaning:

<http://adapt2.sis.pitt.edu/pitt/annotated/annotated-demo/foo?act=pyt3.2&usr=me&grp=001&sid=001>

- adapt.sis.pitt.edu – Server Address
- pitt – Protocol name
- annotated – contentType name

- annotated-demo – contentPackage name
- act – unique name of the example

**Process Flow:**

Retrieve the object from the database → Level 1 parsing → Level 2 parsing → Front-end access

**Installation with Eclipse:**

<https://www.youtube.com/watch?v=OmnKsgm2tjw>