How To Create Your Own OpenPhys Style Site

Introduction

You can create your own website that operates like ‘OpenPhys’. As described below in detail, you will need to:

1. choose a website name (url)
2. find a computer (‘web server’) to run (‘host’) your website
3. download the software files
4. create your lessons, mapping the material to pages in a 2D grid
5. replace the physics content files (text and images) with your own content files (text and images) based on the above lessons
6. Use the tools included with the software files to convert the files into a website (‘build’)
7. upload the website files (‘build’) to your web server

Setup: Get a website name (Domain) and Web Hosting

A domain name refers to the website name (url) used to visit your site, such as http://openphys.med.ualberta.ca/. ‘Web Hosting’ refers to the web server computers where the files of the website are stored and served to the internet from. You can find a more detailed explaination of domain name and hosting at: https://www.youtube.com/watch?v=S4CWFoTZxSA. If you have an IT department, pick a website name for your project then ask them to get the domain and set up hosting. If you do not have an IT department, you will need to investigate the many options for choosing domain names and web hosting and pick what works best for your situation.

Start Project: Get the files (Clone OpenPhys Repository)

All the files needed are freely available on the internet. They are stored at 'github' which is a popular website used for archiving open source software projects. For your own website you will need to copy ('clone') the OpenPhys files (repository) to create your own project. Once you have your own version you can make any changes you need.

If you have an IT department, they can do this step for you. If not, you will need to become familiar with git and github. Luckily, github includes some easy to follow steps at https://github.com/ to help you get started. When ready, copy the OpenPhys files (clone https://github.com/OpenPhysProject/OpenPhys.git) as a base for your own project (repository). Once again, instructions on how to clone a repository can be found at https://help.github.com/articles/cloning-a-repository/.

Update Content: Replace Files With Your Own Work

Most of the following steps can be done by your IT department, but you will need to create the lesson and the related text.

Replace the logo with something appropriate for your project

1. In the app/content/logo folder, replace the svg image files logo-icon.svg (image), logo-tagline.svg(online learning tag), and logo-wordmark.svg (OpenPhys name) with your own graphics.
2. If you do not want one of the 3 aspects of the logo (icon, wordmark, or tagline), you can remove them from app/index.html in <div class="logo> starting at line 19.

Update First Page Text (Intro and Outro Tiles)

1. Find the Intro and Outro tiles in app/index.html in <div class="lesson-view-container"> starting at line 36. They are html div elements labeled with the classes intro-tile and outro-tile.
2. Replace the text inside of the lesson-tile-intro-title and lesson-tile-intro-content. You can include more than one pairing of title and content, but keep in mind that space for this content is limited.
3. Replace the multiple instances of lesson-tile-outro-title and lesson-tile-outro-content with text suitable to your site.
4. intro-tile and outro-tile need to exist as other parts of the project depend on them.
5. Intro and Outro tiles will expand to fit all content included. Keep this in mind when adding content to them, as the site will start to look broken if the tiles get to long.

Remove all existing content files

1. In the app/content folder, remove all the subfolders with lesson in there name (leave Intro and Logo)
2. Remove related lesson script links from app/index.html (starting at line 160, everything after <!-- Content--> except Intro).

Create the lessons you want to share on your website

1. Plan your lesson to map it to pages in a 2D grid layout (see working examples on http://openphys.med.ualberta.ca/).
2. Progress to the right in the 2D grid is meant to be progress through a lesson.
3. Progress down in the 2D grid is meant to represent more difficult concepts that are related to the lesson above.
4. Progress up in the 2D grid is meant to relay less difficult concepts that are related to the lesson below.
5. There should be one main horizontal grouping of pages (primary path) that is meant to be the focus of the lesson.
6. There can be horizontal gaps between pages in the 2D grid.
7. There can be as much or as little content as you want, but consider that a very large 2D map may be overwhelming to new users and be difficult to use. In these cases it might be better to break content into smaller groups.
8. For each page in the 2D map, write the text and collect any other assets (ie images) that will exist on that page.
9. Repeat this process for each lesson you want to create

Create new content files

Your IT depart can do this step for you, but it will be time consuming.

1. Using the lesson you created above, follow the instructions in HowToAddLesson to start adding your own content files.
2. Refer to the various HowTo's in the app/content folder for instructions on how to add specific types of page content, such as:
   a. Custom code to give pages with extra behavior ReadMe
   b. Custom visual styles ReadMe
   c. Quizes ReadMe
3. Repeat this for each lesson.

Create multiple lessons

1. Repeat the 2 steps above (Create Lesson and Create Content Files) to create as many lessons as you want.

Build: Update Your Website With Your New Files

This step can be done by your IT department.

1. Follow the steps in the readme file to set up the software tools needed
2. Create the website files by running grunt build in the command line
3. Upload the files to your hosting (web servers). We use ftp via FileZilla.

Polish: More Advanced Changes

Alter WebSite Base Appearance

1. You can change the base website visuals by altering the scss files in the app/styles folder.
2. The app/styles/settings folder contains scss files that define all the global values used in styles. The primary value of this is to give complete consistency and make it very easy to make application wide changes.
3. The app/styles/views folder can be used to change specific details of different base pages and elements.

Alter Website Base Behavior

1. You can change the base behavior of the site by altering the javascript files in the app/scripts folder.
2. An overview of the code architecture can be found here