

13-08-2019_Glenn-Andre-Patrick-Key_CA1_FP1

What is strongly typed language?

- Strongly typed language checks the type of variable before performing an operation on it. Java is an example of a strongly typed language

What is difference between JavaScript and Java.

- JavaScript is a weakly typed language unlike Java. Although JS (JavaScript) syntax is based on Java, Java is a compiled language that compiles to an intermediate bytecode language making it better to run both on the server side and client side. JS is an interpreted language run through an interpreter on the client side – But with Node.js you can run JavaScript on the server side as well.

Write code to display an if statement in JS.

```
If (1 < 2) {  
    return true  
};
```

Write code to display a single comment.

```
// this is a comment in code.
```

Write out a detailed set of instructions to your friend Jess. She wants to learn how to use the Podcast Section of the website. Remember to include which buttons to click and how everything works. These set of instructions should explain how this player works. Remember that Jess is not very clued up with technology, so you need to give her a proper set of instructions. Please plan out the use of a video player also.

Hi Jess.

- Locate to the top of the webpage, left click with your mouse (tap with one finger if you have a touch pad) on the search bar.

- type in a title /genre or “podcast” of your choice and hit return.

- Now the page should have loaded a list of choices underneath. Click on the orange play button to listen to any podcast or song of your choice or click on the album/podcast picture for more details.

- if you clicked the album picture, press the orange play button to listen. Underneath you will find public comments and ratings and have the choice to like and subscribe.

- On the right-hand side of the page you will find related songs and artists.

- If you wish to return to the last page/song/podcast, press the “←” at the top left of your browser.

Video guide

- Locating to the top of the page underneath the search bar, you will find a button named “videos”. Press this button and your search will list up videos instead of podcasts.
- Click the orange play button or the whole preset image or video title to start streaming the video.
- On the right-hand side of the page you will find related videos and channels.
- If you wish to return to the last page/video, press the “←” at the top left of your browser.

Using the baseline which you have created in question 5. Write out pseudo code that plans how this application will work. Remember planning is imperative. Make use of variables and conditional statements which you learnt about in this course. You can use the buttons for the player as variables. Please use pseudo code to plan out a player for a video player as well.

Audio Player

IF playButton is pressed:

 IF player is paused:

 IF current timer is between a & b:

 THEN play audio from current timer;

 THEN play;

IF pauseButton is pressed:

 IF player is not paused:

 IF current audio is paused between a & b:

 THEN pause at current timer;

 THEN pause;

IF stopButton is pressed:

 THEN stop and reset timer;

IF fastForward is pressed:

 IF player is not paused:

 IF audio timer is between a & b:

 THEN fast forward player from current timer;

 THEN fast forward player;

 ELSE skip to the next song;

IF rewindBack is clicked:

 IF player is not paused:

 THEN rewind player:

 IF timer is between a & b

 THEN rewind from current timer;

 ELSE IF rewind is double clicked:

 IF player is paused:

 THEN play previous song;

 ELSE rewind 30 seconds;

Video Player

IF playbutton is pressed:

 IF video is paused:

 IF current timer is between a & b:

 THEN play video from current timer;

 THEN play;

IF pauseButton is pressed:

 IF video is playing:

 IF current video timer is between a & b:

 THEN pause video at current timer;

 THEN pause video;

IF stopButton is pressed:

 THEN stop video and reset timer;

IF fastForward is pressed:

 IF video is not paused:

 IF video timer is between a & b:

 THEN fast forward 15 seconds from current video timer;

 THEN jump 15 seconds forward:

 ELSE skip to next video;

IF rewindBack is clicked:

 IF video is not paused:

 IF video timer is between a & b:

 THEN rewind video 5 seconds from current timer

 THEN rewind video 5 seconds:

 ELSE rewind 15 seconds;

IF fullScreen button is clicked:

 IF video player is not fullscreen:

 THEN open up window in full screen:

ELSE return to normal video player:

IF volumeButton is hovered:

 THEN userInput adjust slider from 0% – 100%:

 ELSE IF userInput does not adjust slider:

 THEN volumeButton resets to previous setting;

 FOR EACH 1% in slider set the % in volume capacity of video player

ELSE IF volumeButton is clicked:

THEN mute/un-mute volume;

IF subtitlesButton is clicked:

 IF video subtitles are available:

 THEN list different subtitles from a to z:

 FOR EACH subtitle language set the language userInput clicks;

 ELSE ignore action;

IF video quality is clicked

 IF higher quality is available

 THEN list quality ranging from low to high:

 FOR EACH quality choice set quality userInput clicks;

 ELSE IF userInput does not click on any choice from the quality list:

 THEN ignore action;

IF video timer reaches b:

 IF video is playing:

 THEN load relevant videos to choose next;

 ELSE DO nothing;