💯 ASSIGNMENT

Assignment 1

 Cary Jardin · Feb 2 · Notified 39 people

We will review in class, and I will code the assignment in class so don't freak out if you don't know HTML/Javascript!

Language:
HTML - JAVASCRIPT
Some have asked if they can code this in another language... After thinking about it, I am going to say NO.. We need to embrace this as a Programming Languages class, and explore new languages..

How the Assignment will be Graded:
Generally, I will be grading each assignment based on how much effort you put into it. In the comments, tell me if this was your first time with the language and I will adjust accordingly. However, I have been doing this long enough that I have a pretty good understanding of how much effort tasks take and I want you to "geek out" over assignments. Meaning.. Take the assignment, do the base, and then explore!

Completeness:
All submitted assignments must actually work for even partial credit. Additionally, every line of code MUST be documented in a separate file with the extension '.narration' to include the following format

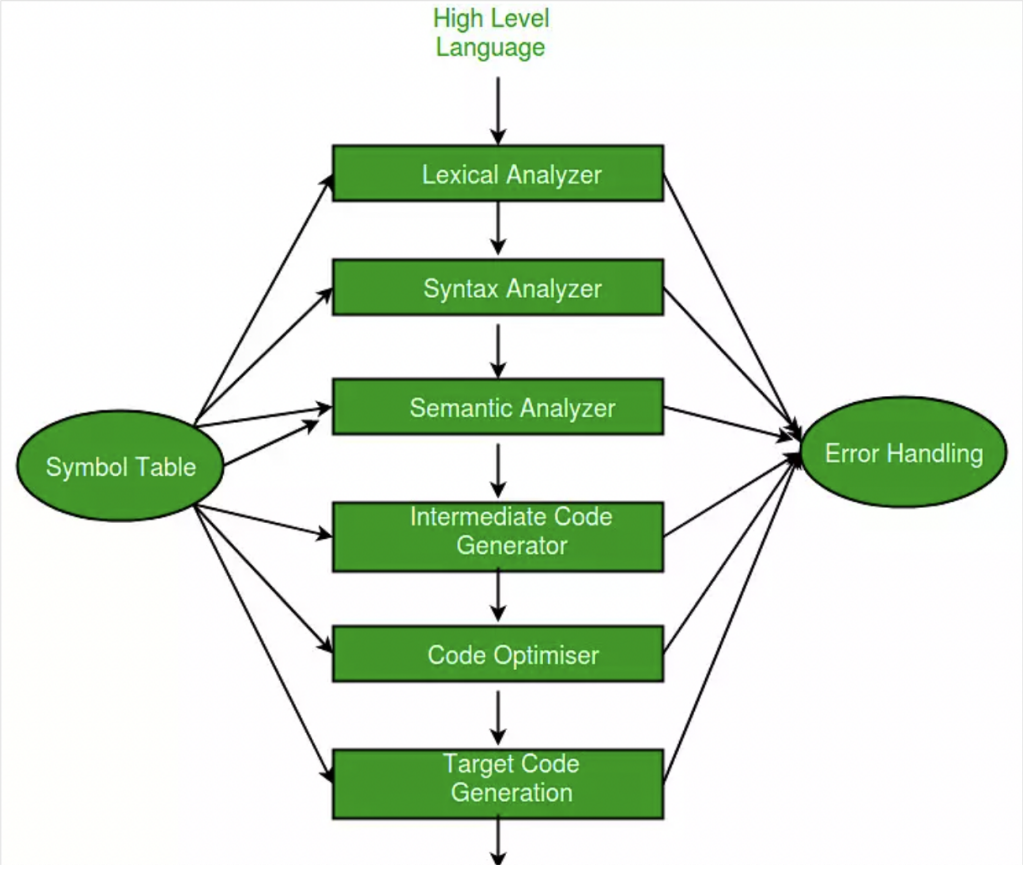
line number : Narration
Example
1 : declare variable to hold sound file

Assignment Due Date:
All assignments are due by February 21st.

Submitting Your Assignment:
Email me a zip file with your code and any supporting files.

Assignment Motivation:

- Write a simple language that can be compiled directly into sound output.
- Experience the difference between Interpreted and Compiled execution.
- Parse a simple language
- Lexical, Syntax, Semantic Analysis with generation of Intermediate Code
- Stretch goal of Code Optimization and Error Handling heuristics



compilerDesign.jpg · 96.1 KB · View full-size · Download

The Assignment:

Create an HTML Text Area that allows user input of coded "notes"; a "note" can be any single character you choose.

A
  B
    C
Write a script to read in the contents of the text area into string. Loop through each line of the string in such a way that each line may be accessed:

line.charAt(0); // A
next line
line.charAt(1); // B
next line
line.charAt(2); // C

Place each parsed "Note" into an array that would look like:

notes = ["A", "B", "C"]

Once the intermediate "notes" are compiled, loop through each note and play a sound associated with the "note."

new Audio('a.mp3').play()

You will need to figure out how to NOT have all the "Notes" play at the same time.. Ideas:

- Use audioElement.addEventListener('ended', function() {}) and chain the notes together

- Chain the "play" with setTimeout(function(){ }, 3000);

Use whatever mp3s you want, but "notes" on different lines should not play at the same time.

THEN...

In a second html text area, compile the notes into a stand alone HTML page using token substitution.

<textarea id="compiled_code" rows="20" cols="40">

</textarea>

```
var htmlTEMPLATE = "<!doctype html>"
htmlTEMPLATE = htmlTEMPLATE + "<html><head><\/head ><body>"
htmlTEMPLATE = htmlTEMPLATE + "<script>@@@PLAY_CODE <\/script>"
htmlTEMPLATE = htmlTEMPLATE + "<\/body ><\/html>"
```

From the video, the primary presented "interpreting loop" looked like this:

```
  timing = 0
  index_ms_offset = 1000
  for (index = 0; index < notes_parsed.length; index++) {
    if( notes_parsed[index] == 'A'){
      setTimeout( function(){
                  new
Audio('zapsplat_animals_cat_kitten_meow_004_30180.mp3').play()
                }, index_ms_offset * index )
    }else if (notes_parsed[index] == 'B' ){
      setTimeout( function(){
                  new
```

```
                    new
Audio('zapsplat_cartoon_rise_upwards_futuristic_002_44571.mp3').play()
                }, index_ms_offset * index )
        }
    }
```

This will actually play each sound.. However, we are in a Programing Languages class that you also need to learn compiler theory... So.. the next step is code generation. Just like the loop above, except we are generating JavaScript..

```
    code_output = ""
    timing = 0
    index_ms_offset = 1000
    for (index = 0; index < notes_parsed.length; index++) {
        if( notes_parsed[index] == 'A'){
            code_output = code_output + "setTimeout( function(){"
            code_output = code_output + "new
Audio('zapsplat_animals_cat_kitten_meow_004_30180.mp3').play()"
            code_output = code_output + "}, " + index_ms_offset + " * " + index + ");
"
        }else if (notes_parsed[index] == 'B' ){
            code_output = code_output + "setTimeout( function(){"
            code_output = code_output + "new
Audio('zapsplat_cartoon_rise_upwards_futuristic_002_44571.mp3').play()"
            code_output = code_output + "}, " + index_ms_offset + " * " + index +");
"
        }
    }
```

Notes you are NOT PLAYING THE SOUNDS.. You are creating a string that has Javascript code in it.. When done:

```
$("#compiled_code").val(htmlTEMPLATE.replace("@@@PLAY_CODE",
code_output))
```

My Output:

> <!doctype html><html><head></head ><body><script>setTimeout(function() {new Audio('zapsplat_cartoon_rise_upwards_futuristic_002_44571.mp3').play()}, 1000 * 0); setTimeout(function(){new Audio('zapsplat_animals_cat_kitten_meow_004_30180.mp3').play()}, 1000 * 1); </script></body ></html>

If I cut and paste "My Output" into a new file.. It plays the "Compiled" notes... the output is a fully functional, stand alone HTML page..

AGAIN.. the goal is for you to generate javascript that is then pushed into a new, stand alone HTML document... This is a Compiler!

Stretch Goal Ideas.. These are JUST IDEAS.. We will be adding on to this project in later assignments:

- Play two files at once
- Add a number in-front of the note for how many times it should play in a row
- Add a decimal in-front of the note for how fast the note should play

Have fun with it!