CHIL: CSS HTML Integrated Language

Isaac White - iaw2105 **Annania Melaku -** amm2324 **Ami Kumar -** ak3284 **Gil Chen-Zion -** gc2466

Motivation

More and more technology is shifting towards the web. Many people want to create functional and attractive web sites but don't necessarily have a working knowledge of all the different programming styles that make up a web page. Between HTML and CSS, there are complexities and common practices that the new user has no knowledge of that would greatly improved on the look and feel of their sites. Needing to have an understanding of both HTML and CSS before starting development hinders the barrier for entry in creating web pages.

Applications

Web development is currently isolated in technique from many other programming languages because it is often strictly not programming. As a result, the main web technologies for constructing a web page (HTML, CSS) do not really function as a typical programming language. Our language resolves this division by providing a standardized programming interface for the creation of styled HTML documents, without needing to learn syntactical oddities in the CSS specification for applying similar properties to different types of elements. It also provides the ability to programmatically loop through different types of element creation, or define more complex elements made up of basic types provided by the language.

Language Specifications

Data Types:

Key Word	**Description**
int	non-decimal number
String	sequence of characters
element	fundamental container that can contain text or images
style	characteristics of an element
elements	any fundamental element in html DOM tree

Reserved Words:

Keyword	Description
hlevel	ranges from 1-6
url	the path to a file or image, requires parentheses
background	the fill of the implied element
arrangement	layout format for elements
font	style of printed characters
outerSpacing	margin
innerSpacing	padding
width	int or string declaring x-size of object
function	declares a function
children	array of elements contained within an element
return	passes back calculated result (std)

Functions:

Function Name	Description
push	adds object to array (std)
remove	takes properties out of style for object

Function declaration:

```
function name(parameters) {
        ...
        return someElement;
}
```

All functions return elements. The element to be returned is indicated by the `return` keyword.

Operators:

Function Name	Description
=	assignment operator
+, -, *, /	basic arithmetic operators
<, <=, >, >=, ==	comparison operators, by value
++,--	increment and decrement
%	modulus
“”	denotes a string

Comments:

```
/* This is a multi-line
       comment */

// This is a single-line comment
```

Comments can be nested.

Sample Program:

```
function drawGrid(int elementCount, String columnWidth, String filePrefix) {
       element grid = “”;
       grid.children.arrangement = “inline”;
       grid.children.width = columnWidth;
       for (int i = 0; i < elementCount; i++) {
       element currentImage = “”;
       currentImage.background = url(filePrefix +  i + “.jpg”);
       grid.addChild(currentImage);

       return grid;
}
element firstHeader = “This is a test page”;

firstHeader.hlevel = 1;
firstHeader.style = {
       background = “white”;
       innerSpacing = 4;
       outerSpacing = 5;
}
firstHeader.comment = “Here’s a simple title for our page.”;
```

```
style bigParagrah = firstHeader.style;
bigParagraph.remove(outerSpacing,innerSpacing);
bigParagraph.font = “Georgia”;

Page.elements.push(firstHeader);
//(Page.elements[0] = firstHeader)

element checkItOut = “Look at all these pretty pictures!”;
checkItOut.comment = “This is a public comment rendered by CHIL.”;

checkItOut.style = bigParagraph;
Page.elements.push(checkItOut);

element firstGrid = drawGrid(16,”25%,””);
firstGrid.children.innerSpacing = 4;
element secondGrid = drawGrid(8,”50%”,”someFolderName/image”);

Page.elements.push(firstGrid);
Page.elements.push(secondGrid);
```

Output: http://www.columbia.edu/~iaw2105/plt/