# Dinosaur Substitution

This packet will walk you through a Chrome Extension that reads the text in the HTML on a webpage and substitutes the word "lizard" for the word "dinosaur."

## Core Concepts

* content\_scripts
* recursive Javascript functions
* switch statements
* string manipulation

## Source Files

*manifest.json*

The notable part of this manifest is the "content\_scripts" section which describes which Javascript files to run for what pages.

The "matches" item tells Chrome to run our script for all pages. Look up regular expressions or "regexes" if you want to learn more about matching strings. The "js" item lists only our substitution.js script. The "run\_at" item tells Chrome to run the script after it has already rendered the page's HTML.

{  
 "manifest\_version": 2,  
 "name": "Lizard to Dinosaur",  
 "version": "1.0",  
 "description": "Replaces the text 'lizard' with 'dinosaur.'",  
 "content\_scripts":   
 [  
 {  
 "matches": ["\*://\*/\*"],  
 "js": ["substitution.js"],  
 "run\_at": "document\_end"  
 }  
 ]  
}

*substitution.js*

This script is executed as defined in the manifest file.

// This is the entry point for the script. It calls the walk function on the  
// HTML body of the loaded webpage.  
walk(document.body);  
  
// This variable keeps track of whether or not we've already alerted the user.  
var haveAlerted = false;  
  
// defining the walk function which takes a "node" as the only argument.  
// node is an HTML element  
function walk(node)   
{  
 // declaring the variables child and next, without assigning values to them.  
 var child, next;  
   
 // a switch statement  
 switch (node.nodeType)  
 {  
 case 1: // Element  
 case 9: // Document  
 case 11: // Document fragment  
 // This block of code is executed if the nodeType is any of the three above  
 // Get the top element of the node  
 child = node.firstChild;  
 // This block will keep looping until the variable child has no value (is null)  
 while (child)  
 {  
 // call this function on the child. Notice that the walk function  
 // is now calling itself. This is called recursion.  
 walk(child);  
 // get the next element on the node, which is the child's first sibling  
 next = child.nextSibling;  
 child = next;  
 }  
 // Exit the switch  
 break;  
   
 case 3: // Text node  
 // Perform the substitution on this node because it is a text node.  
 handleText(node);  
 break;  
 }  
}  
  
// This function performs the actual text substitution on the textNode argument  
function handleText(textNode)   
{  
 // The nodeValue attribute is the text that is inside the node.  
 var v = textNode.nodeValue;  
   
 v = v.replace(/\bLizard\b/g, "Dinosaur");  
 v = v.replace(/\blizard\b/g, "dinosaur");  
 v = v.replace(/\bLizards\b/g, "Dinosaurs");  
 v = v.replace(/\blizards\b/g, "dinosaurs");  
  
 // check if dinosaurs appear in the page and if we have not already alerted the user  
 if ((v.indexOf("dinosaur") != -1 || v.indexOf("Dinosaur") != -1) && !haveAlerted) {  
 // show a pop-up to the user  
 alert("WARNING! This page contains dinosaurs!");  
 // change the haveAlerted variable to true so we don't show the error again  
 haveAlerted = true;  
 }  
   
 // set the value of the text node to be the text that we've edited.  
 textNode.nodeValue = v;  
}

## Running your Chrome extension

You should load this extension as usual from chrome://extensions, but you won't need to do anything to launch it. Just navigate in Chrome to a page that contains the word "lizard" and see what happens!

For example, try http://en.wikipedia.org/wiki/Lizard

## See what you can do!

* Only substitute words in h1 tags.
* Make an extension that changes the word "Soccer" to "Football".