# Number Guesser

This Chrome Application will ask the user to guess a secret number that it generated randomly. It will tell the user whether their guesses are too low, too high, or just right!

## Core Concepts

* variables
* Math in Javascript
* if/else statements
* CSS

## Getting Started

As usual, create a new folder for this packet and put the following files inside.

## Source Files

*manifest.json*

{  
 "name": "Number Guesser",  
 "description": "This app tries to get you to guess the secret number.",  
 "version": "1.0",  
 "manifest\_version": 2,  
  
 "app": {  
 "launch": {  
 "local\_path": "guess.html"  
 }  
 }  
}

*guess.html*

<!DOCTYPE html>  
<html>  
<head>  
 <script src="guess.js" type="text/javascript"></script>  
 <!-- This tells the browser to look at the file guess.css to determine the style of the page. -->  
 <link rel="stylesheet" type="text/css" href="guess.css">  
</head>  
<body>  
 <h1 id=guess-title>Guess my number!</h1>  
  
 <!-- An unordered list ("ul") of list items ("li"s), currently empty. -->  
 <ul id='guess-list'>  
 </ul>  
  
 <form id='guess-form'>  
 <input type="text" name="guess" id="guess-input">  
 </form>  
</body>  
</html>

*guess.js*

document.addEventListener('DOMContentLoaded', function() {  
  
 // The secretNumber variable refers to a random number between 0 and 10,000.  
 // Math.random returns a random value betweeon 0.0 and 1.0  
 // Math.floor takes a decimal number and truncates the decimals, leaving an integer  
 var secretNumber = Math.floor(Math.random()\*10000);  
 // numGuesses will keep track of how many guesses the user has made. So far, none.  
 var numGuesses = 0;  
  
 // variables referencing elements of the HTML  
 var guessList = document.getElementById('guess-list');  
 var form = document.getElementById('guess-form');  
 var guessInput = document.getElementById('guess-input');  
  
 // This is the code we execute each time the form is submitted (when the user  
 // hits enter in the input box).  
 form.addEventListener('submit', function(event) {  
 event.preventDefault();  
  
 // the user's input  
 var guess = guessInput.value;  
 // increase the number of guesses.  
 numGuesses += 1;  
  
 // a new list item  
 var li = document.createElement('li');  
 // sets the text of the list item to the user's input  
 li.textContent = guess;  
 // adds the list item to the unordered list guessList to display to the user.  
 guessList.appendChild(li);  
  
 // another new list item  
 var li2 = document.createElement('li');  
 // this conditional checks if the user's input is a number  
 if (isNaN(parseFloat(guess))) {  
 li2.textContent = "You must guess an integer.";  
 // change the color of this list item to red  
 li2.style.color = "red";  
 } else if (guess == secretNumber) {  
 li2.textContent = "You guessed it! It took you " + numGuesses + " guesses.";  
 // to green...  
 li2.style.color = "green";  
 } else if (guess < secretNumber) {  
 li2.textContent = "Tooooo looow...";  
 // to blue...  
 li2.style.color = "blue";  
 } else {  
 li2.textContent = "Too high!";  
 // to yellow.  
 li2.style.color = "yellow";  
 }  
 // add this list item to the guessList also.  
 guessList.appendChild(li2);  
  
 // always display the bottom of the list and clear the input form  
 guessList.scrollTop = guessList.scrollHeight;  
 guessInput.value = '';  
 });  
});

*guess.css*

/\* This block changes the font for the entire HTML body. \*/  
body {  
 font-family:"Helvetica Neue", Arial, sans-serif;  
}  
  
/\* Defines the styling for the guess-list. \*/  
#guess-list {  
 height: 400px;  
 border: 1px solid #ccc;  
 /\* Automatically create a scrollbar if the content is taller than the window.\*/  
 overflow-y: auto;  
 padding: 15px;  
}  
  
#guess-list li {  
 /\* The default style for a list item is a bullet, so here we set it to none. \*/  
 list-style-type: none;  
 padding: 0;  
 /\* Set the font size. \*/  
 font-size: 16pt;  
}  
  
input[type=text] {  
 /\* Makes the input box the width of the entire browser window. \*/  
 width: 100%;  
}

## See what you can do!

* Report how many times the user guesses too high and too low.
* Don't count guesses when the user didn't input a number.