

Search Bars

- Input Fields
- Parsing Input
- Toggling Elements

Input Fields

Creating a default input field in HTML:

```
<input class=searchbar type="text" placeholder="Search For a Project.." >
```

Input Fields

Type	Value
button	defines a clickable button
checkbox	defines a checkbox
color	defines a colorpicker
file	defines a file-select field and 'Browse' button
image	defines a formatted file-select field for images
email / password	defines a formatted text field for email / passwords
text	defines a text field

Types are browser-dependent.

A color picker will look different in IE than in Chrome...

- ~~Input Fields~~

- Parsing Input

- Toggling Elements

Parsing Input

```
$(document).ready(function(){  
  
    $(".searchbar").keyup(function(event) {  
  
        var k = $(this).val();  
  
        loadData(k);  
  
    });  
});
```

keyup() is called whenever a key is pressed and passes the resulting string to loadData()

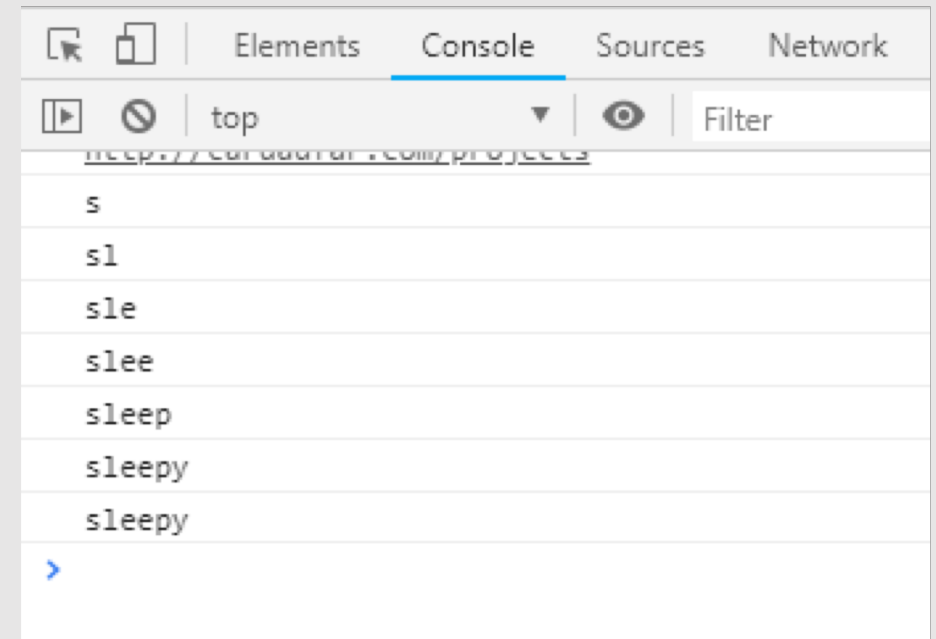
Parsing Input

```
$(document).ready(function(){  
  
    $(".searchbar").keyup(function(event) {  
  
        var k = $(this).val();  
  
        console.log(k);  
  
    });  
});
```

What would happen if I typed 'sleepy' into the searchbar?

Parsing Input

```
$(document).ready(function(){  
  
  $(".searchbar").keyup(function(event) {  
  
    var k = $(this).val();  
  
    console.log(k);  
  
  });  
});
```



Every keypress triggers the event handler. Eager execution.

- ~~Input Fields~~

- ~~Parsing Input~~

- **Toggle Elements**

Toggling Elements

Assuming all HTML elements stored in `toggleElems`, we can write a simple search-toggle algorithm:

```
function loadData( str ) {  
  
    var param = str.toLowerCase();  
    param = '.' + param;  
  
    for(var i = 0; i < toggleElems.length; i++) {  
        $(toggleElems[i]).css('display', 'none');  
    }  
  
    $(param).addClass('display', 'inline-block');  
  
}
```

} Turn all elements off

} Turn `param` element on

Toggling Elements

```
<div class=family >  
  <div class=red ></div>  
  <div class=green ></div>  
  <div class=blue ></div>  
  <div class=yellow ></div>  
</div>
```

```
function loadData( str ) {  
  
  var param = str.toLowerCase();  
  param = '.' + param;  
  
  $('.'+param).children().css('display', 'none');  
  $(param).addClass('display', 'inline-block');  
  
}
```

Assuming each div is a square with the color corresponding to its name (except family), what would result in calling loadData('green')?

Homework Ideas

- ❑ Insert input fields such as searchbars in HTML.
- ❑ Have JQuery receive and log the output. Verify this by inspecting elements.
- ❑ Include a toggle button that shows certain divs when clicked.

Live Demo