

Let's build an  
**autocomplete**

Building an autocomplete  
might **look easy**

There are several  
**important details**

# Feature requirements

# Feature requirements

- Look as designed
- Type and show a list of options
- Click an option and select it

SEARCH

SEARCH

Ams|

Amsterdam

Amstel

Amsterdam Central Station

Schiphol Airport (AMS)

SEARCH

SEARCH

Amsterdam|

Amsterdam

Amstel

Amsterdam Central Station

Schiphol Airport (AMS)

Missing requirements

# Missing requirements

- Accessibility
- Cross-browser compatibility



Accessibility

Screen readers hints

# Screen reader hints

```
<label for="autocomplete-1">Fruit:</label>
<input
  id="autocomplete-1"
  role="combobox"
  aria-autocomplete="list"
  aria-activedelement="autocomplete-1-item-0"
  aria-expanded="false"
>
<ul>
  <li id="autocomplete-1-item-0">Apple</li>
</ul>
```

# Screen reader hints

```
<label for="autocomplete-1">Fruit:</label>
<input
  id="autocomplete-1"
  role="combobox"
  aria-autocomplete="list"
  aria-activedelement="autocomplete-1-item-0"
  aria-expanded="false"
>
<ul>
  <li id="autocomplete-1-item-0">Apple</li>
</ul>
```

# Screen reader hints

```
<label for="autocomplete-1">Fruit:</label>
<input
  id="autocomplete-1"
  role="combobox"
  aria-autocomplete="list"
  aria-activedelement="autocomplete-1-item-0"
  aria-expanded="false"
>
<ul>
  <li id="autocomplete-1-item-0">Apple</li>
</ul>
```

# Screen reader hints

```
<label for="autocomplete-1">Fruit:</label>
<input
  id="autocomplete-1"
  role="combobox"
  aria-autocomplete="list"
  aria-activedelement="autocomplete-1-item-0"
  aria-expanded="false"
>
<ul>
  <li id="autocomplete-1-item-0">Apple</li>
</ul>
```

# Screen reader hints

```
<label for="autocomplete-1">Fruit:</label>
<input
  id="autocomplete-1"
  role="combobox"
  aria-autocomplete="list"
  aria-activedelement="autocomplete-1-item-0"
  aria-expanded="false"
>
<ul>
  <li id="autocomplete-1-item-0">Apple</li>
</ul>
```

⌘ + Tab



Keyboard behavior

# Keyboard behavior

- Up and down arrows navigates
- Tab while one item is highlighted selects the item
- Enter selects the item without submitting the form
- Escape restores original item

⌘ + Tab

And also

- User agent should be able to increase font sizes
- Visually differentiated items should have enough contrast

⌘ + Tab

Cross-browser  
compatible

Different keyboard events

**input** event isn't available everywhere

```
const eventName = 'oninput' in document.documentElement  
  ? 'input' : 'keydown';  
element.addEventListener(eventName, handleInput);
```



Browser autofills

Specific quirks

# Safari autocorrect conflicts with Voice Over

```
<input  
  autocomplete='off'  
  autocorrect='off'  
  spellcheck='false'  
>
```

# The Checklist

# User Interface

- The user interface is familiar
- Items are properly highlighted as options are being browsed
- Enough contrast is provided for users

# Behavior

- Keyboard navigation and option selection are allowed
- Up/down arrows, tab and return keys work as expected
- Pasting content triggers the autocomplete

# Accessibility

- Users are able to increase the base font size
- The input element must have an appropriate label
- ARIA attributes are set to properly hint screen readers

# Robust

- Things appear in the correct position and z-index, and disappear when they should
- It works with events from multiple browsers
- Hacks might not work on future updates, new browsers or platforms



Takeaways

# Takeaways

- Reimplementing native controls requires **more than basic** interactions
- Always leverage **already existing** native controls
- Consider **the checklist**

# Thank you!

[wesley.so](https://wesley.so)