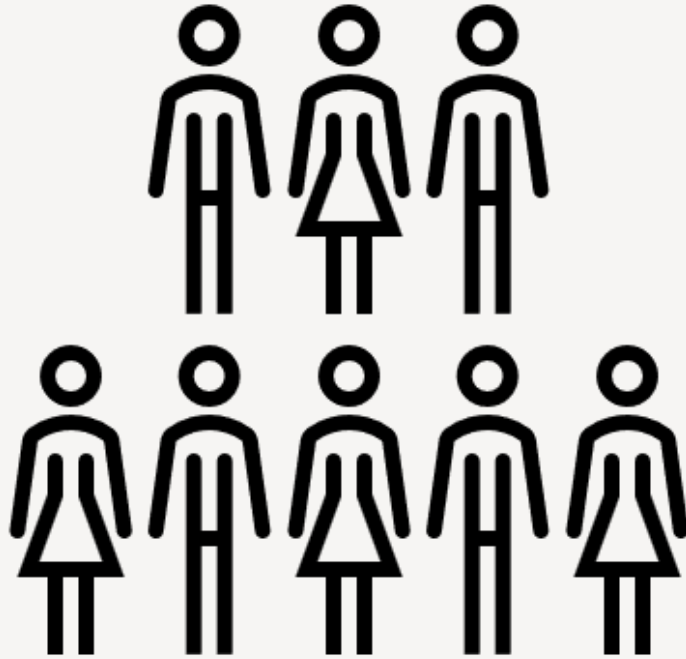


# Bag of spanners

State of the Browser



# Everyone's responsibility



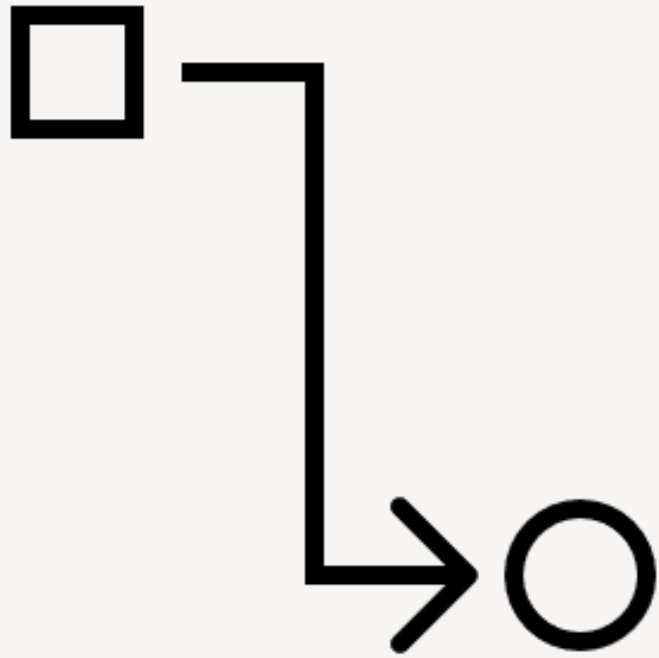


## Part of your toolkit

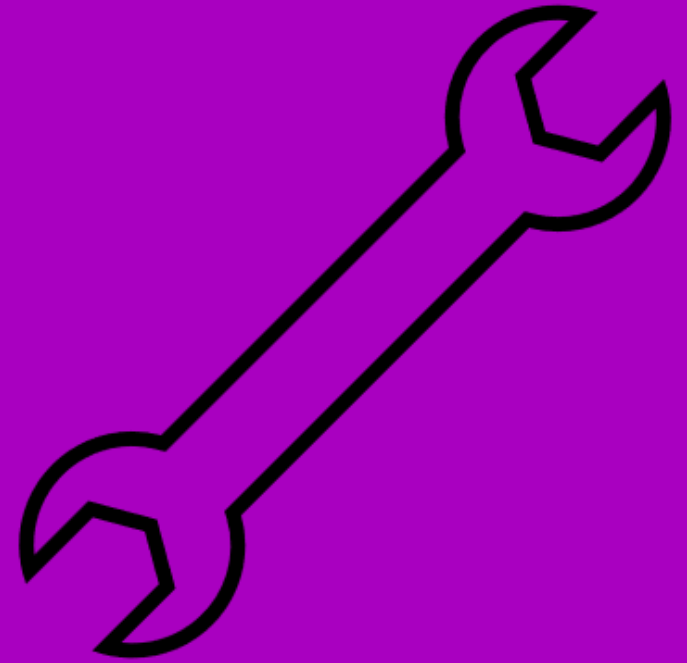




## Cause and effect



# Mechanics





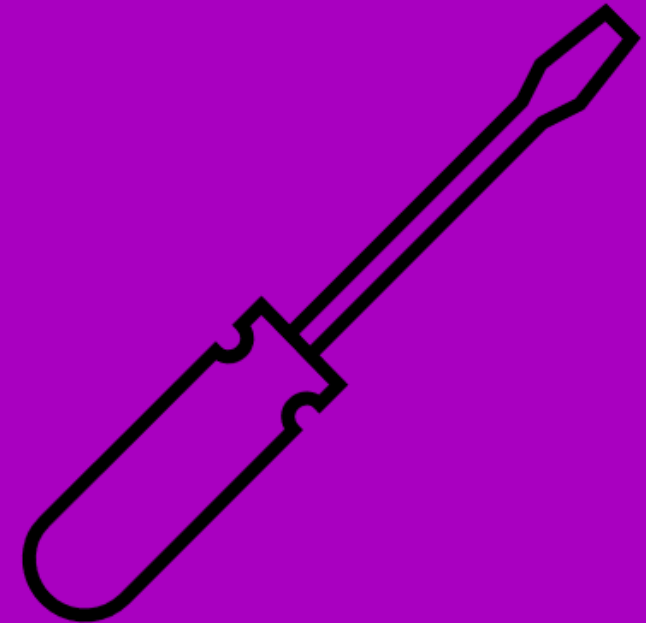
# Platform accessibility API

- MSAA + Iaccessible
- MSAA + IAccessible2
- NSAccessibility Protocol
- AT-SPI
- UIAutomation

# Accessibility tree

Role	Name	Checks
▼ document:	"Tink - Léonie Watson - On technology, food & life in the digital age"	▼ Checks No checks for this node.
▶ link:	"Skip to content"	▼ Properties
▶ landmark:	""	name: "Light"
▶ landmark:	""	role: "toggle button"
▼ section:	""	▼ actions: [...]
▶ landmark:	"secondary"	0: "Press"
▼ text container:	""	length: 1
▶ heading:	"Theme"	value: ""
▶ toggle button:	"Light"	DOMNode: button.button ⚙
▶ toggle button:	"Dark"	description: ""
▶ landmark:	""	keyboardShortcut: ""
		childCount: 1
		indexInParent: 1
		▼ states: [...]
		0: "pressed"
		1: "focusable"
		2: "opaque"
		3: "enabled"
		4: "sensitive"
		length: 5
		▶ relations: {...}
		▶ attributes: {...}

# Semantics







# Implicit semantics

HTML elements and attributes have implied semantics



# Explicit semantics

Explicit semantics can be applied using ARIA



## Role

An element's role describes its purpose



## The `<button>` element

```
<button></button>
```



## Name

An element's accessible name identifies it



## The `<button>` element

```
<button>Show password</button>
```



# State

An element's state describes its current condition



## The aria-pressed attribute

```
<button aria-pressed="true">Show password</button>
```





## Nav

```
<nav>...</nav>
```



## Nav

```
<nav aria-label="Website">...</nav>
```



# Lists

```
<ul>
```

```
<li>Role</li>
```

```
<li>Name</li>
```

```
<li>State</li>
```

```
</ul>
```



# Navigation

```
<nav aria-label="Website">
```

```
<ul>
```



# Navigation

```
<nav aria-label="Website">
```

```
<ul>
```



# Radio buttons

```
<label for="r1">
```

```
<input type="radio" id="r1" name="colour">
```



# Radio groups

```
<fieldset>
```

```
<legend>Choose a colour</legend>
```

```
<label for="r1">
```

```
<input type="radio" id="r1" name="colour">
```

```
Purple</label>
```

```
<label for="r2">
```

```
<input type="radio" id="r2" name="colour">
```

```
Red</label>
```

```
</fieldset>
```



# Data tables

```
<table>
  <caption>Average daily tea and coffee consumption</caption>
  <tr>
    <th>Person</th><th>Coffee</th><th>Tea</th>
  </tr>
  <tr>
    <th>Njoki</th><td>5 cups</td><td>0 cups</td>
  </tr>
  <tr>
    <th>Ilesha</th><td>1 cup</td><td>2 cups</td>
  </tr>
  <tr>
    <th>Léonie</th><td>0 cups</td><td>25 cups</td>
  </tr>
</table>
```



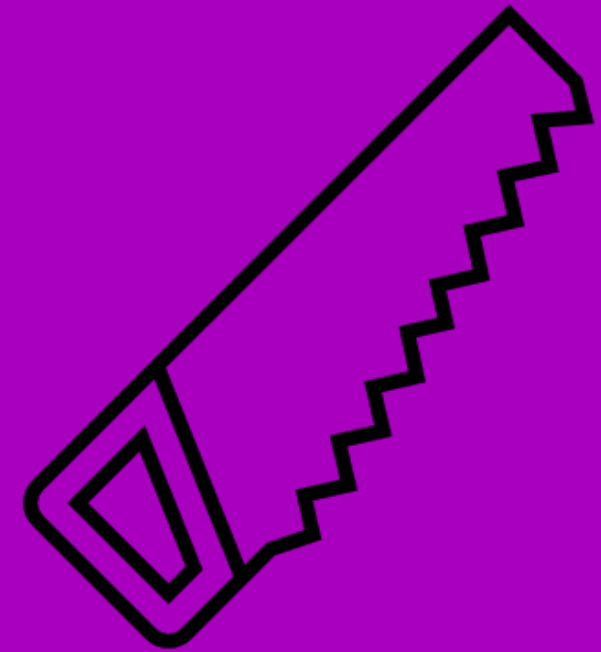
## Demo: HTML data table

The demo data table used in the post on [how screen readers navigate data tables](#) and the associated [screen reader demonstration](#).

Average daily tea and coffee consumption

Person	Coffee	Tea
Njoki	5 cups	0 cups
Iesha	1 cup	2 cups
Léonie	0 cups	25 cups

**Interaction**





# Implicit keyboard support

HTML interactives have implicit keyboard support



# Explicit keyboard support

Explicit keyboard support must be provided for custom HTML



## The `<button>` element

```
<button id="button">Play</button>
```

```
button = document.querySelector("#button");
```

```
button.addEventListener("click", playThis);
```



## The `<a>` element

```
<a href="#" id="button">Play</a>
```

```
button = document.querySelector("#button");
```

```
button.addEventListener("click", playThis);
```

```
button.addEventListener("keydown", function(e) {
```

```
    if (e.keyCode ==13) {
```

```
        playThis();
```

```
    }
```

```
});
```



## The `<a>` element

```
<a id="button">Play</a>
```

```
button = document.querySelector("#button");
```

```
button.addEventListener("click", playThis);
```

```
button.addEventListener("keydown", function(e) {
```

```
    if (e.keyCode === 13 || e.keyCode === 32) {
```

```
        playThis();
```

```
    }
```

```
});
```

```
Button.setAttribute("tabindex", "0");
```



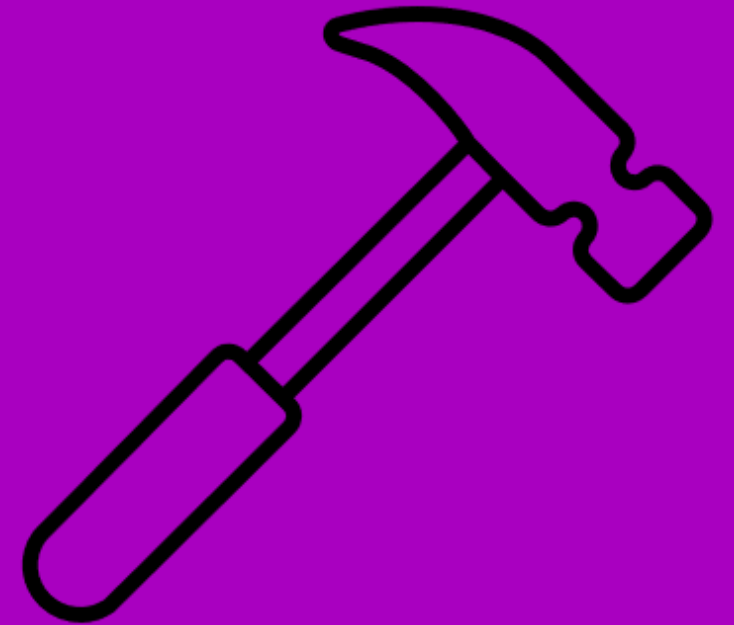
## The <a> element

```
<a id="button">Play</a>
```

```
button = document.querySelector("#button");  
button.addEventListener("click", playThis);  
button.addEventListener("keydown", function(e) {  
    if (e.keyCode == 13 || e.keyCode == 32) {  
        playThis();  
    }  
});  
button.setAttribute("tabindex", "0");  
button.setAttribute("role", "button");
```



**Construction**





# Menubar: Basic HTML

```
<ul>
```

```
<li>
```

```
<a href="#">Categories</a>
```

```
<ul>
```

```
<li> <a href="code.html">Code things</a> </li>
```

```
<li> <a href="life.html">Web life</a> </li>
```

```
...
```

```
</ul>
```

```
</li>
```

```
...
```

```
</ul>
```



# Menubar: Roles

```
<ul role="menubar">
```

```
<li role="none">
```

```
<a role="menuitem" href="#">Categories</a>
```

```
<ul role="menu">
```

```
<li role="none">
```

```
<a role="menuitem" href="code.html">Code things</a>
```

```
</li>
```

```
<li role="none">
```

```
<a role="menuitem" href="life.html">Web life</a>
```

```
</li>
```

```
...
```



# Menubar: Names

```
<ul aria-label="Blog taxonomy">
  <li role="none">
    <a role="menuitem" href="#">Categories</a>
    <ul role="menu" aria-label="Categories">
      <li role="none">
        <a role="menuitem" href="code.html">Code things</a>
      </li>

      <li role="none">
        <a role="menuitem" href="life.html">Web life</a>
      </li>
    </ul>
  </li>
  ...
</ul>
```



# Menubar: States

```
<ul aria-label="Blog taxonomy">
  <li role="none">
    <a role="menuitem" aria-expanded="false" href="#">Categories</a>
    <ul role="menu" aria-label="Categories">
      <li role="none">
        <a role="menuitem" href="code.html">Code things</a>
      </li>
      <li role="none">
        <a role="menuitem" href="life.html">Web life</a>
      </li>
      ...
    </ul>
  </li>
</ul>
```

# Demo: Menu

Categories Tags



# Keyboard commands

- Enter/Space
  - Opens/closes the menu
- Up/Down/Left/Right
  - Cycles through the menus
- Escape
  - Closes the open menu and returns focus to the parent menu

# Demo: Menu

Categories Tags





# Mechanics Semantics Interaction Construction





**Thank you!**

**State of the Browser @LeonieWatson**