



Array Methods Visualised

Array Methods Visualised (part 1)

<code>[●●●●].push(●)</code>	→	<code>[●●●●●]</code>
<code>[●●●●].unshift(●)</code>	→	<code>[●●●●●]</code>
<code>[●●●●].pop()</code>	→	<code>[●●●]</code>
<code>[●●●●].shift()</code>	→	<code>[●●●]</code>
<code>[●●●●].filter(●)</code>	→	<code>[●●]</code>
<code>[●●●●].map((●)=>●)</code>	→	<code>[●●●●]</code>
<code>[●●●●].join("-")</code>	→	<code>"●-●-●-●"</code>
<code>[●●].concat([●●])</code>	→	<code>[●●●●]</code>
<code>[●●[●●]].flat()</code>	→	<code>[●●●●]</code>
<code>[●●●●].slice(1, 3)</code>	→	<code>[●●]</code>



David Mráz
@davidm_ai





Array Methods Visualised

Array Methods Visualised (part 2)

<code>[orange, purple, red, green].some(item => item === red)</code>	→ True	At least one item satisfies the condition
<code>[orange, purple, red, green].every(item => item === red)</code>	→ False	Every item satisfies the condition
<code>[orange, purple, red, green].find(item => item === green)</code>	→ green	
<code>[0, 1, 2, 3].findIndex(item => item === green)</code>	→ 3	
<code>[0, 1, 2, 3].indexOf(red)</code>	→ 2	
<code>[orange, purple, red, green].includes(red)</code>	→ True	
<code>[0, 1, 2, 3].at(2)</code>	→ red	
<code>[orange, purple, red, purple].sort()</code>	→ [orange, purple, purple, red]	Method updates original array
<code>[orange, purple, red, green].reverse()</code>	→ [green, red, purple, orange]	Method updates original array
<code>[orange, purple, red, purple].fill(1, green)</code>	→ [orange, green, green, green]	
<code>[orange, red, green].splice(1, 0, purple)</code>	→ [orange, purple, red, green]	Method updates original array

Legend
 Method updates original array
 Item in the array one by one



David Mráz
@davidm_ai





Array Methods Visualised

Array Methods Visualised (part 3)

Legend:

- Item in the array one by one
- Callback Function
- Method updates original array

[0 1 2 3].findLastIndex() => () == () → 3

[0 1 2 3].findLast() => () == () → 3

[0 1 2 3].lastIndexOf() → 3

[0 1 2 3].toSorted() → [0 1 2 3]

[0 1 2 3].toReversed() → [3 2 1 0]

[0 1 2].toSpliced(1, 0,) → [0 1 2]

[0 1 2 3].copyWithin(0, 2, 4) → [0 1 2 3]

[0 1 2].with(1,) → [0 1 2]

Array.from("0 1 2") → ["0", "1", "2"]

Array.isArray("[0, 1, 2, 3]") → False

Array.of(0, 1, 2) → [0, 1, 2]



David Mráz
@davidm_ai

