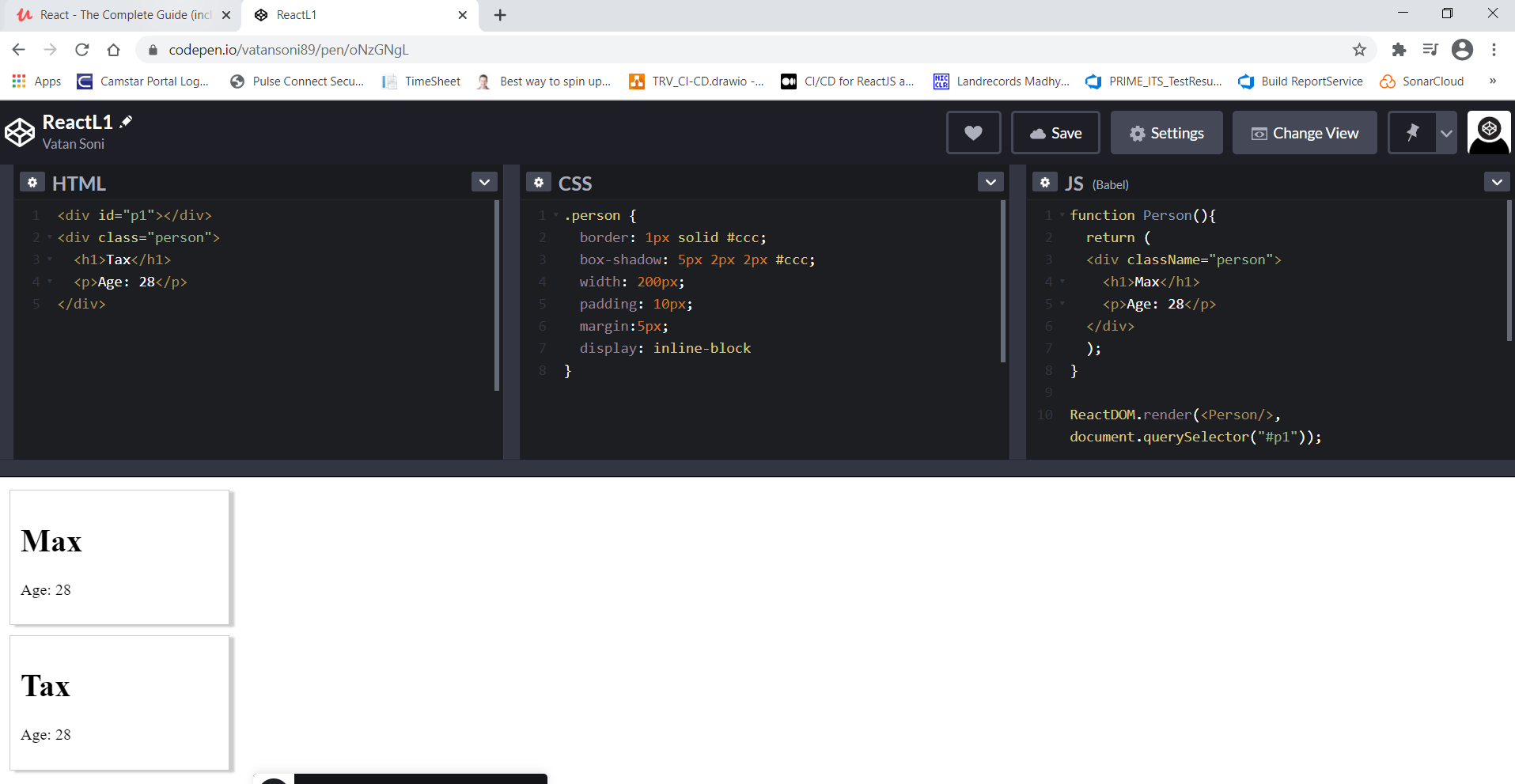
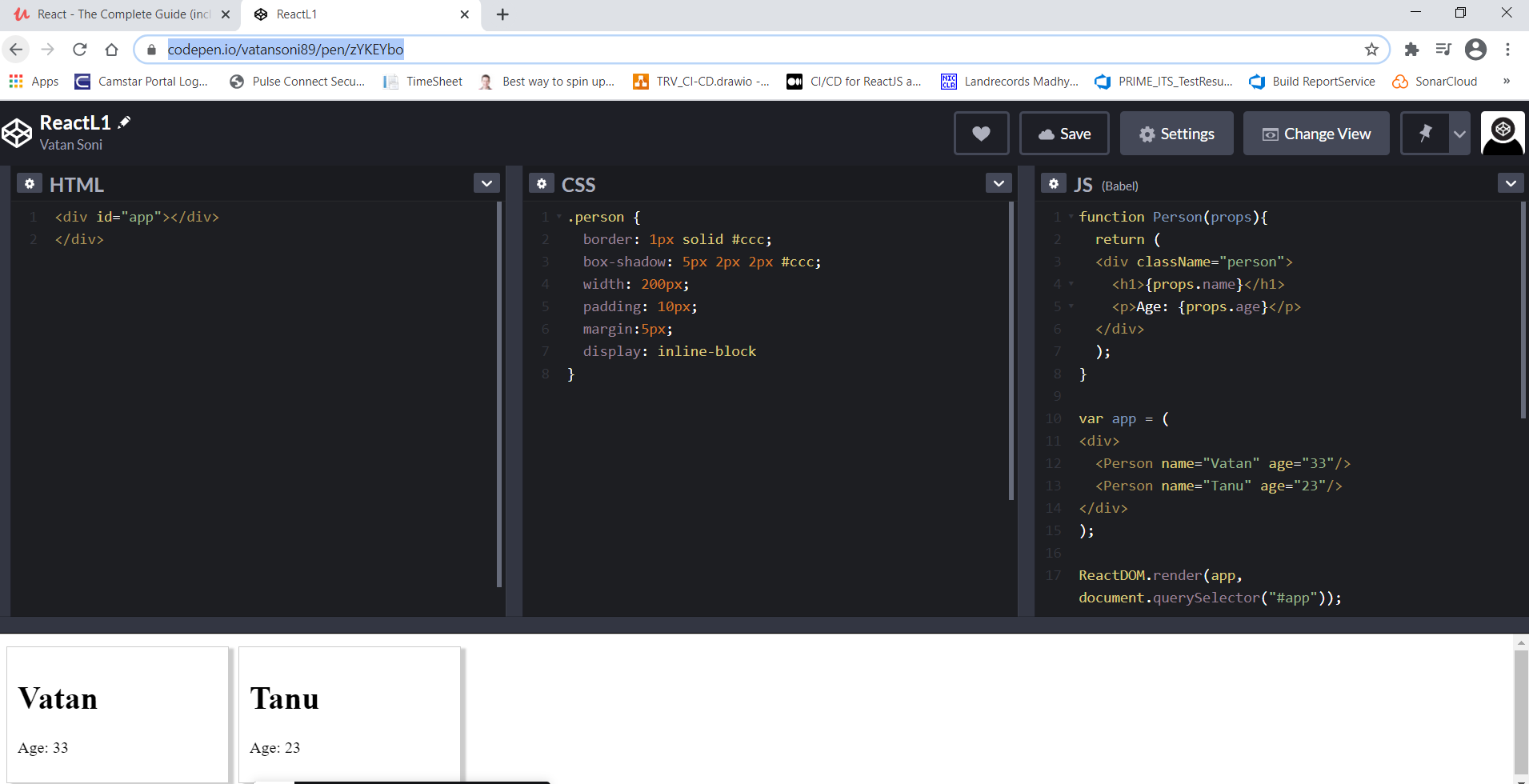
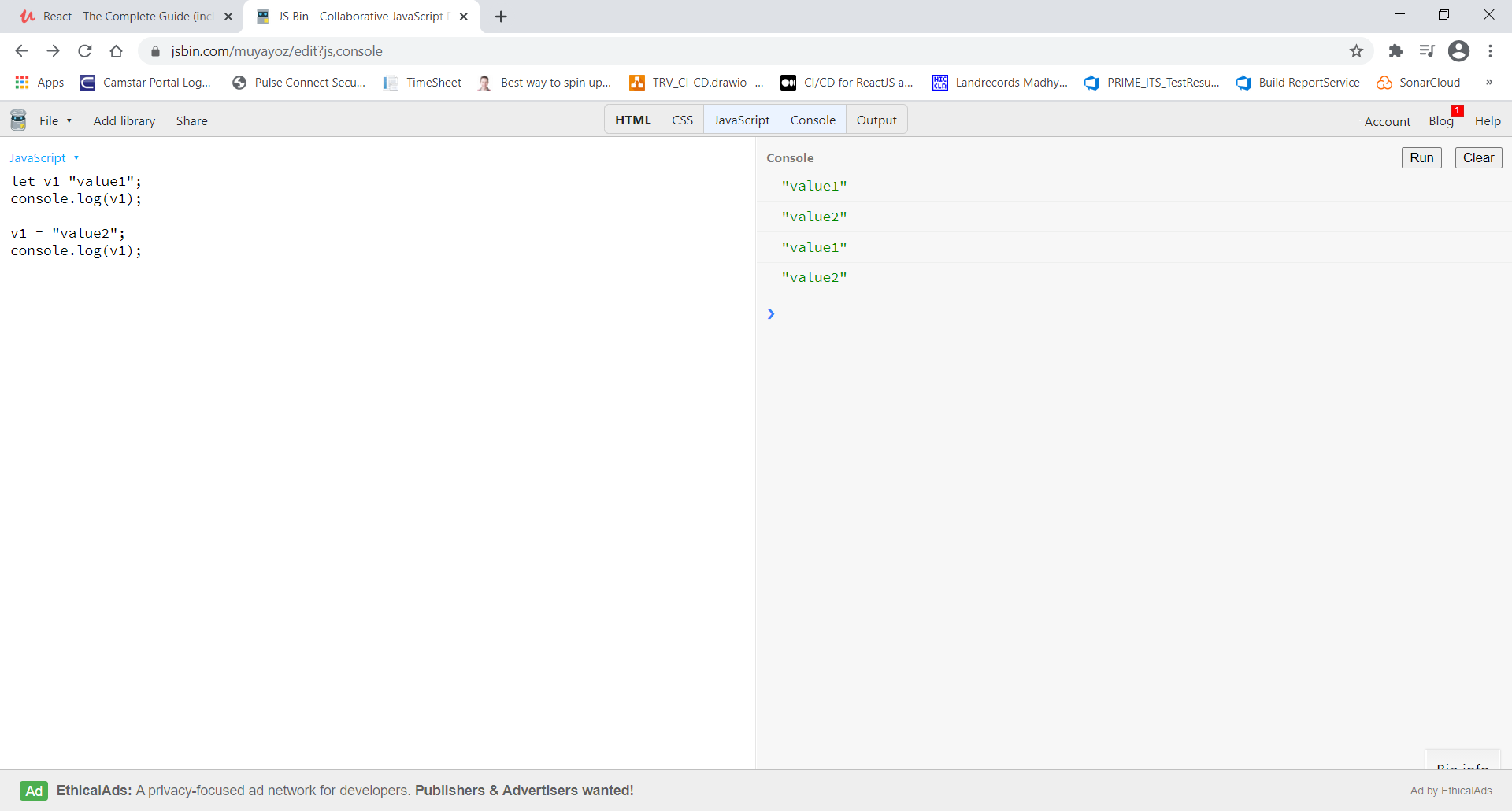
https://codepen.io/vatansoni89/pen/oNzGNgL



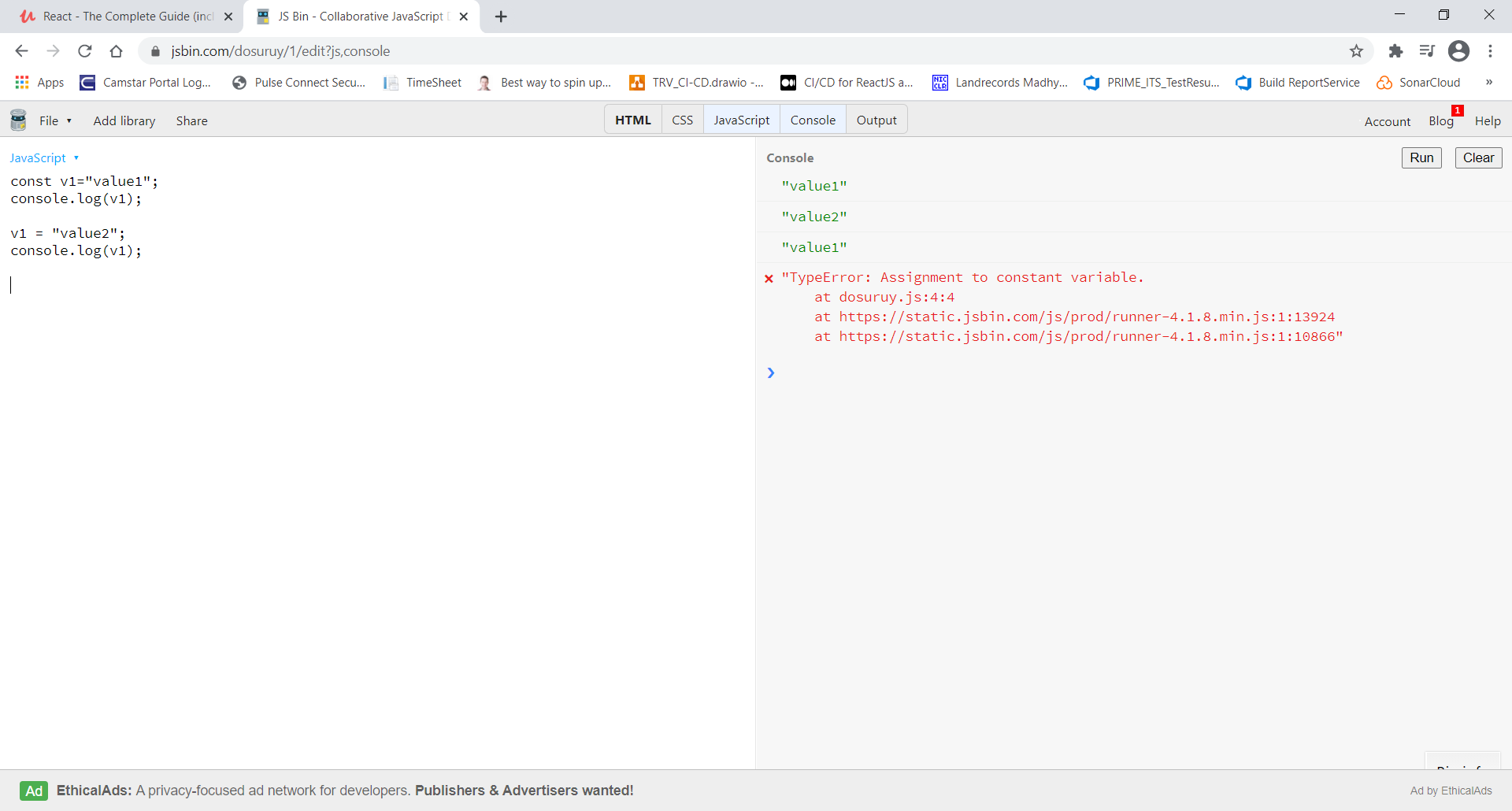
<https://codepen.io/vatansoni89/pen/zYKEYbo>



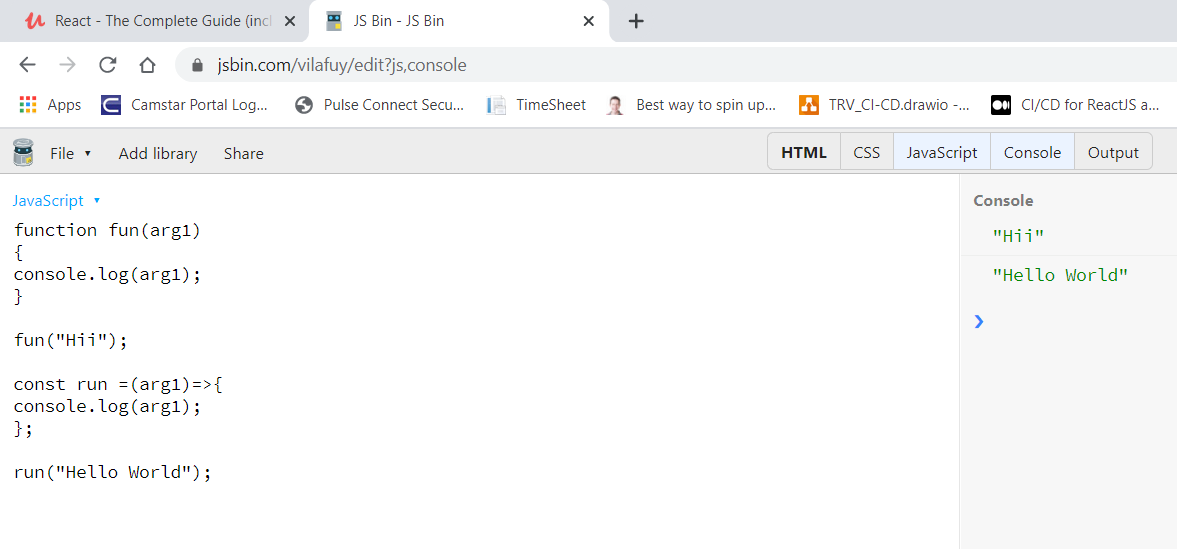
<https://jsbin.com/muyayoz/edit?js,console>



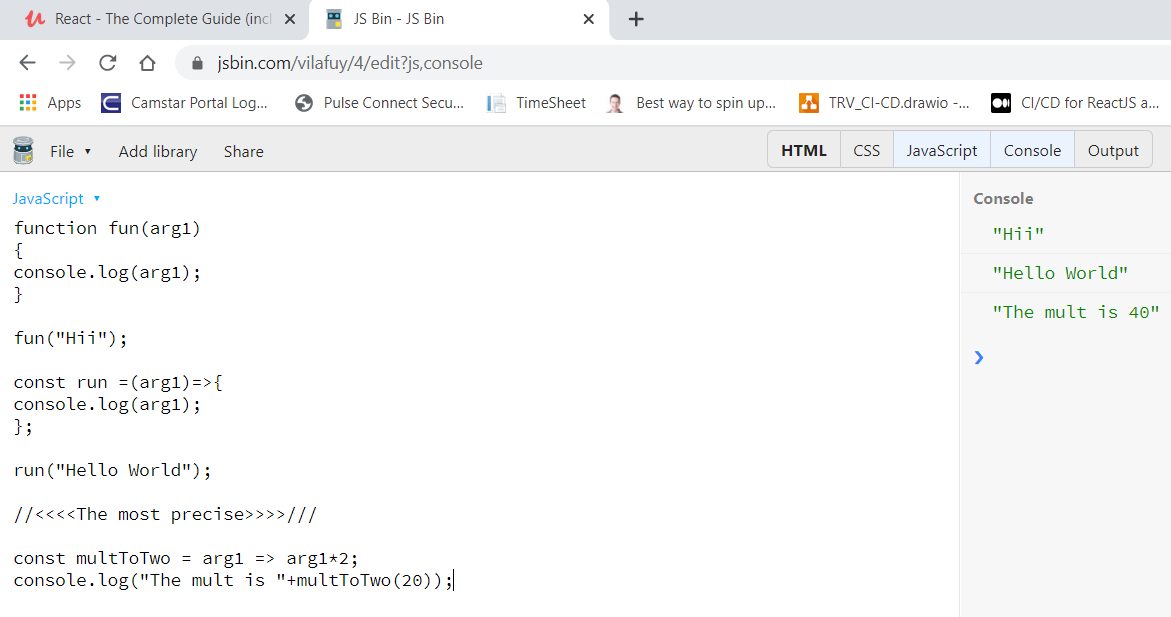
<https://jsbin.com/dosuruy/1/edit?js,console>



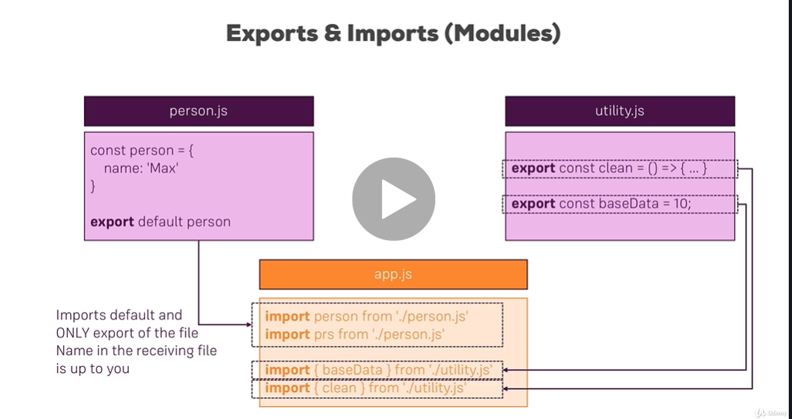
<https://jsbin.com/vilafuy/edit?js,console>

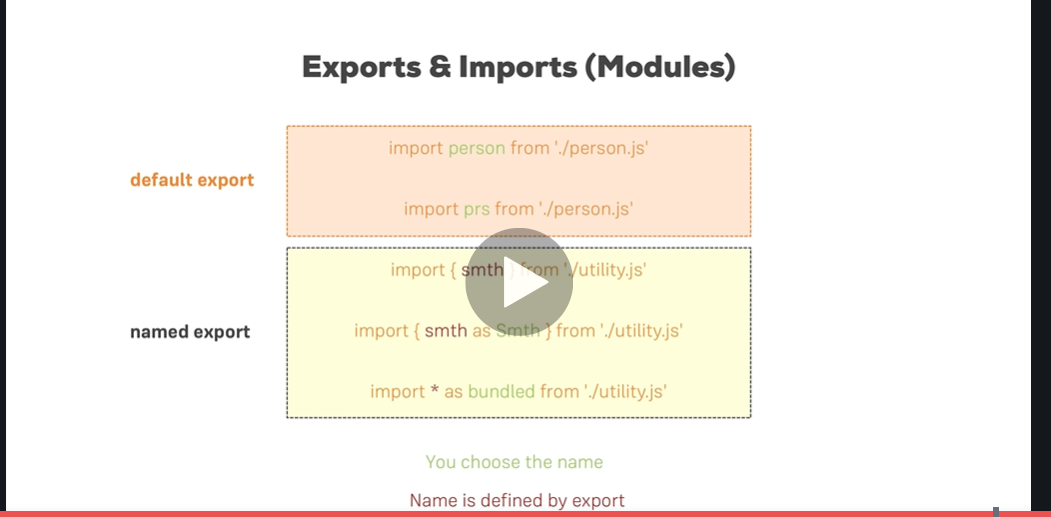


https://jsbin.com/vilafuy/edit?js,console



**Import & Export**

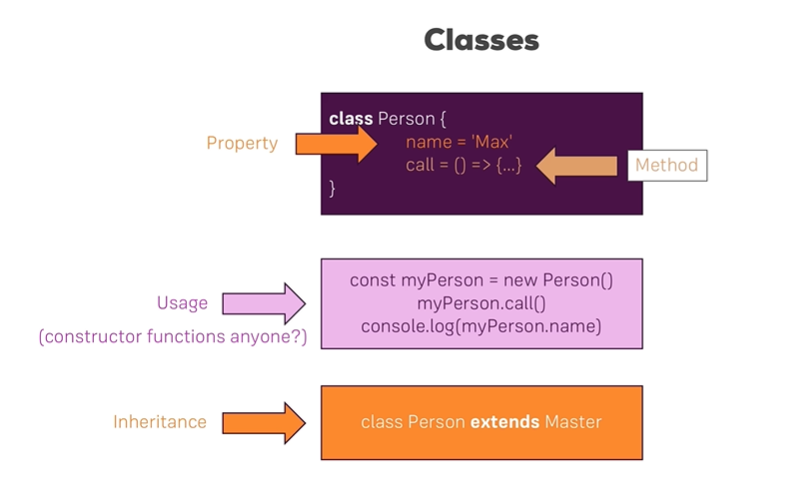




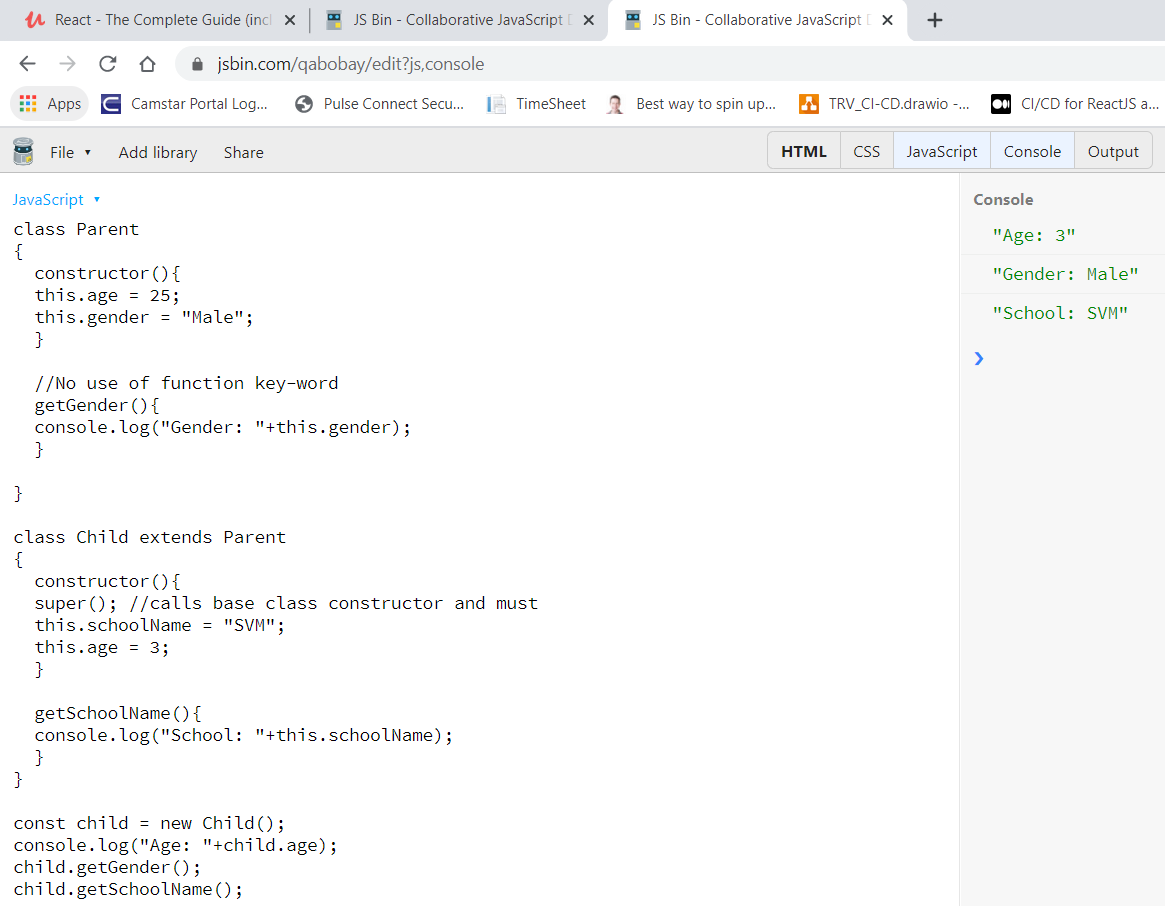
-----------------------------------------------------------------------------------------------------------------------------------------

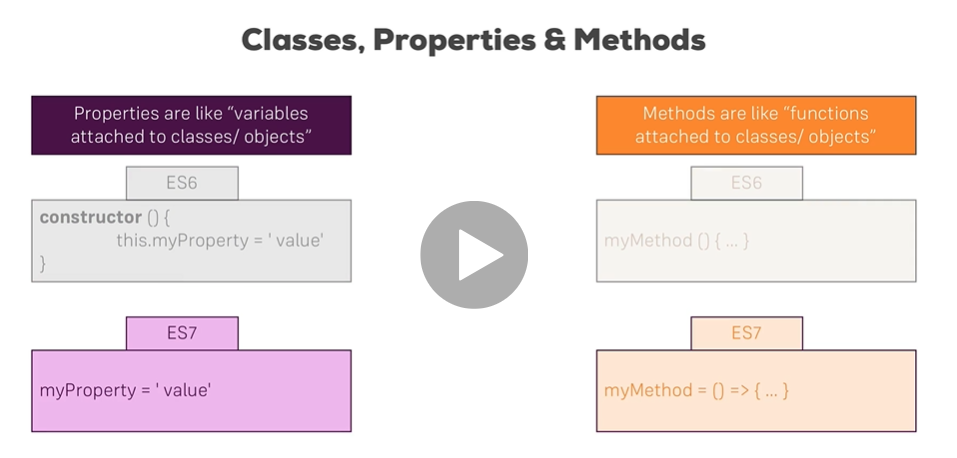
Classes:

https://jsbin.com/qabobay/6/edit?js,console



<https://jsbin.com/qabobay/22/edit?js,console>

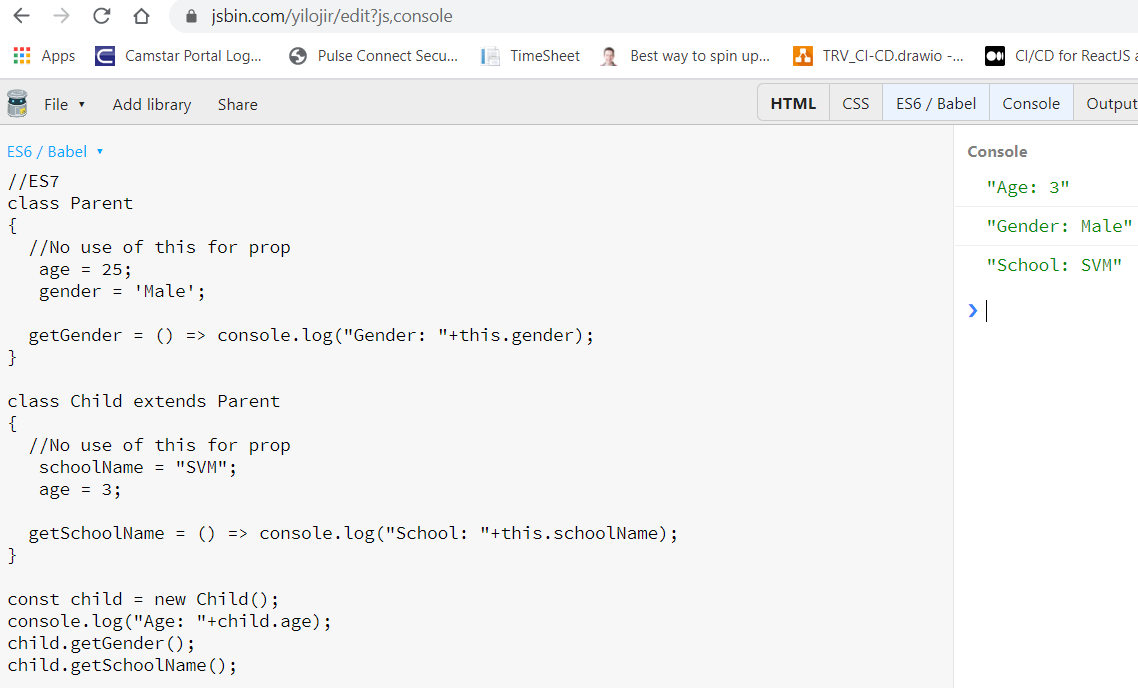




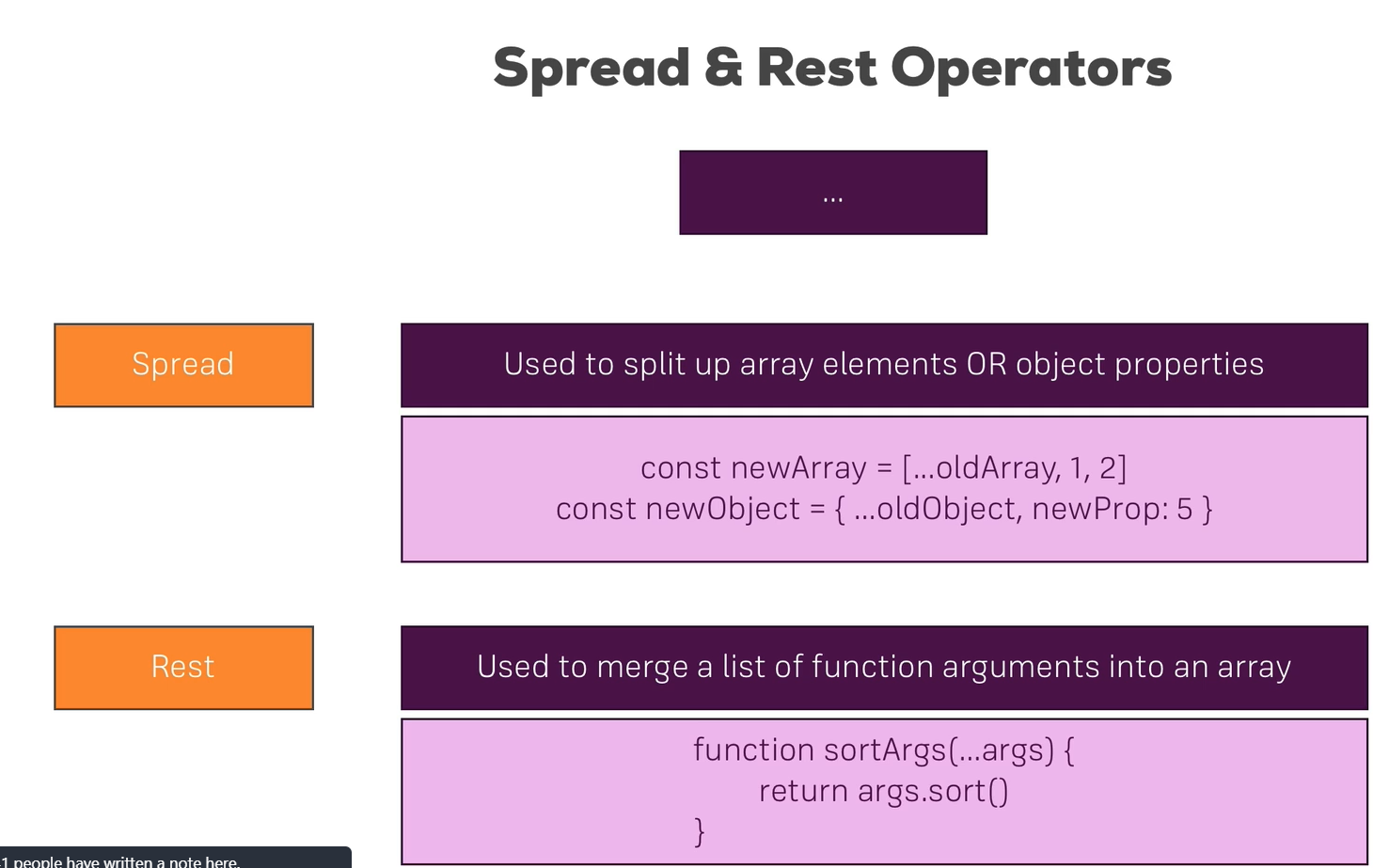
-----------------------------------------------------------------------------------------------------------------------------------------

ES7 class and prop and function:

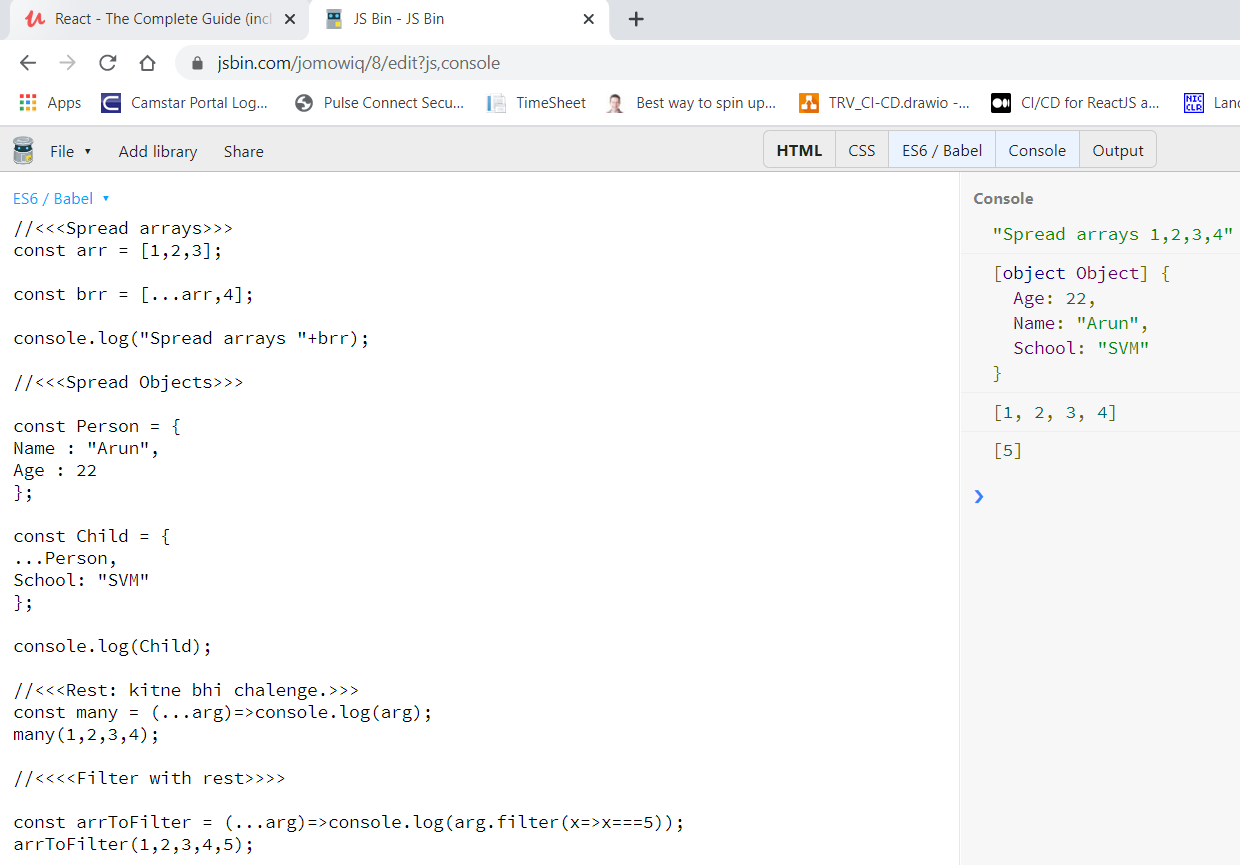
<https://jsbin.com/yilojir/6/edit?js,console>



Spread & Rest:

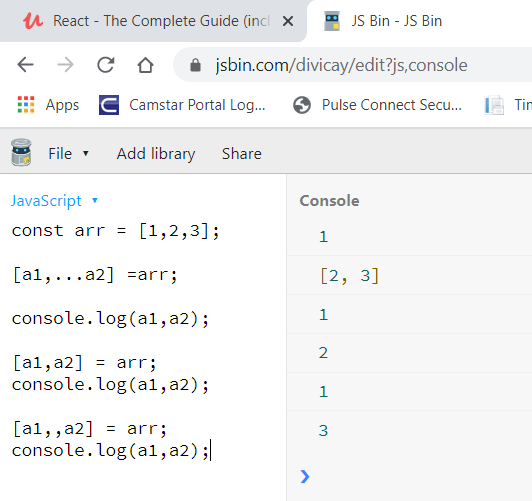


https://jsbin.com/jomowiq/9/edit?js,console



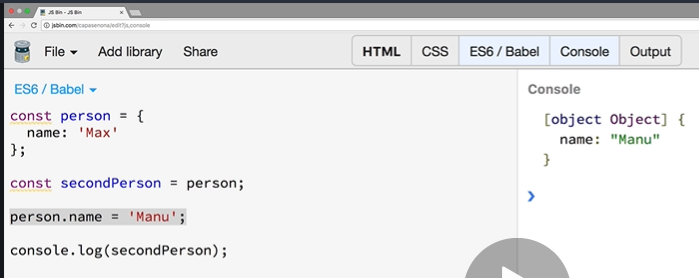
Destructuring:

<https://jsbin.com/divicay/1/edit?js,console>



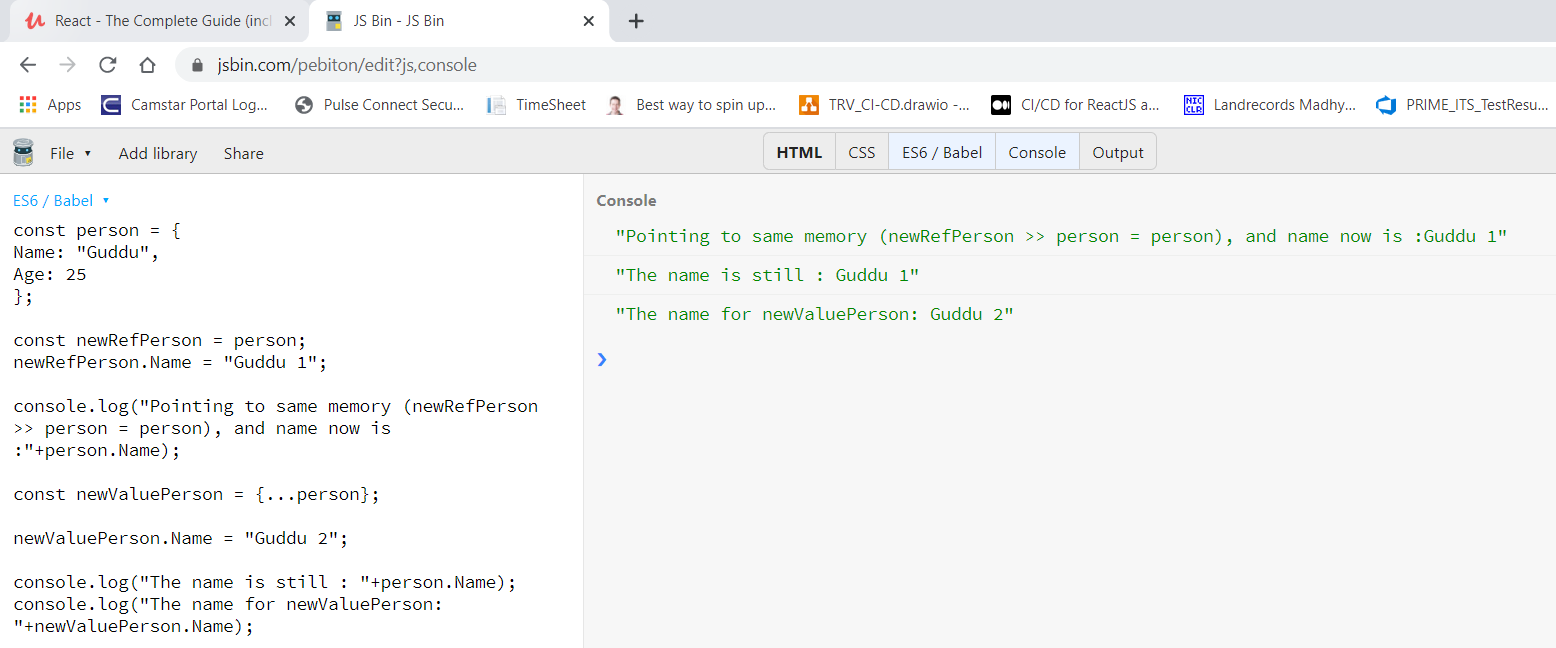
Value & Ref type:

**\***Obj and arrays are ref types.



Copy Ref. type as value type by distructor:

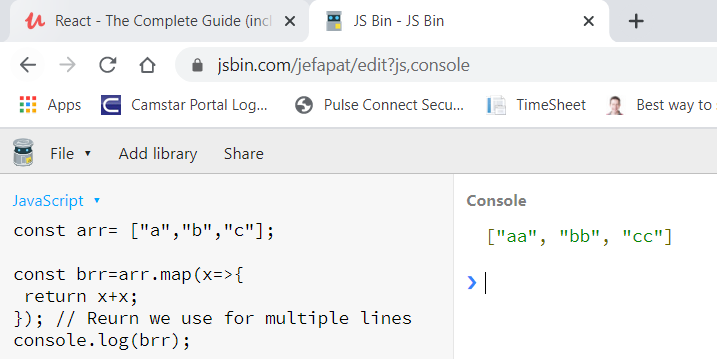
<https://jsbin.com/pebiton/2/edit?js,console>



map():

<https://jsbin.com/jefapat/edit?js,console>

// Reurn we use for multiple lines



**JS Array functions:**

JS Array Functions

Not really next-gen JavaScript, but also important: JavaScript array functions like map() , filter() , reduce()  etc.

You'll see me use them quite a bit since a lot of React concepts rely on working with arrays (in immutable ways).

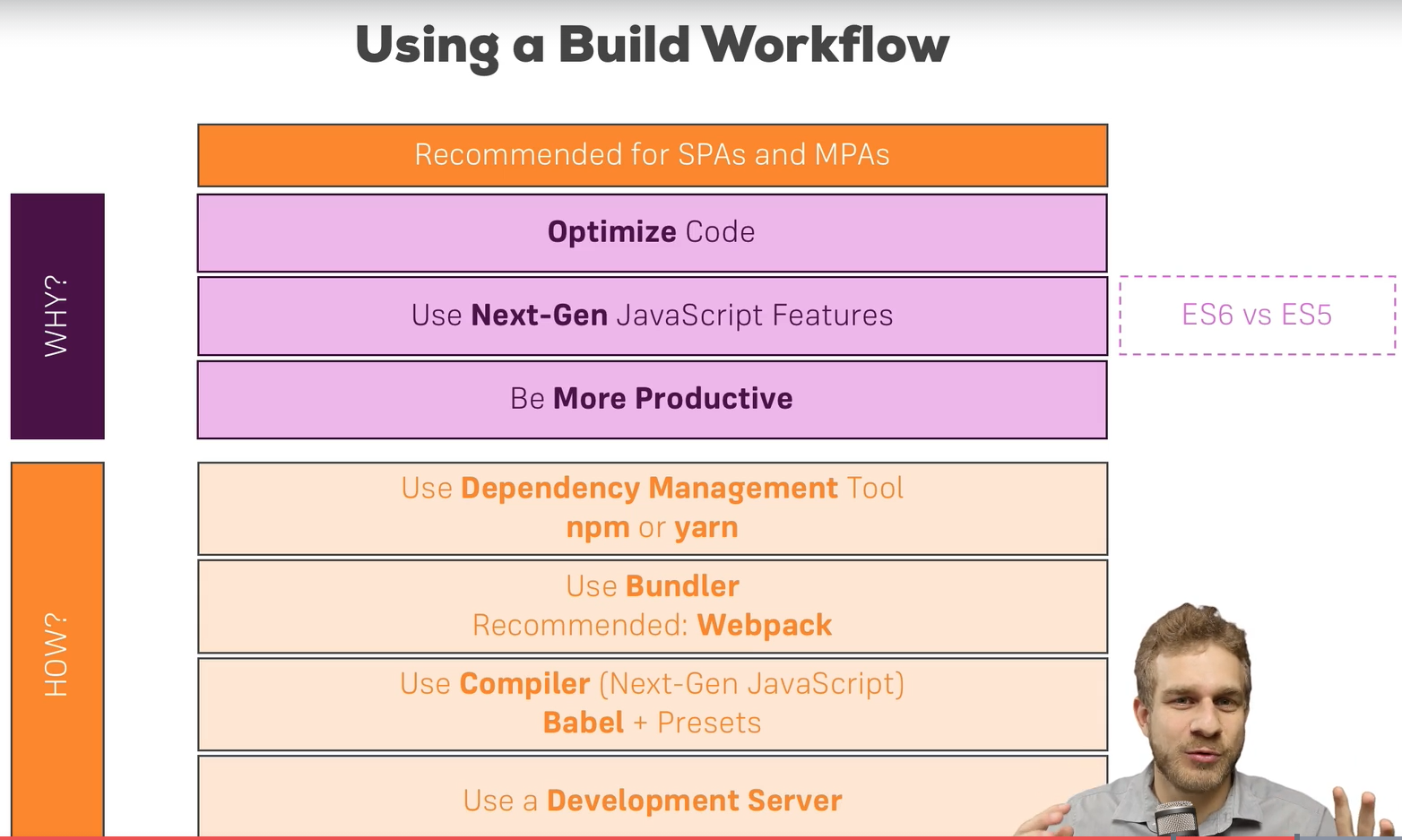
The following page gives a good overview over the various methods you can use on the array prototype - feel free to click through them and refresh your knowledge as required: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array>

Particularly important in this course are:

* map()  => <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map>
* find()  => <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/find>
* findIndex()  => <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/findIndex>
* filter()  => <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter>
* reduce()  => <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/Reduce?v=b>
* concat()  => <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/concat?v=b>
* slice()  => <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/slice>
* splice()  => <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/splice>

---------------------------------------------<<<<<<<Part 2 Done>>>>>>>-------------------------------------------------------

Build workflow



To Which Events Can You Listen?

In the last lecture, we saw that you can react to the onClick event - but to which other events can you listen? You can find a list of supported events here: <https://reactjs.org/docs/events.html#supported-events>

#### Clipboard Events

Event names:

1. onCopy onCut onPaste

Properties:

1. DOMDataTransfer clipboardData

Composition Events

Event names:

1. onCompositionEnd onCompositionStart onCompositionUpdate

Properties:

1. string data

Keyboard Events

Event names:

1. onKeyDown onKeyPress onKeyUp

Properties:

1. boolean altKey
2. number charCode
3. boolean ctrlKey
4. boolean getModifierState(key)
5. string key
6. number keyCode
7. string locale
8. number location
9. boolean metaKey
10. boolean repeat
11. boolean shiftKey
12. number which

Focus Events

Event names:

1. onFocus onBlur

These focus events work on all elements in the React DOM, not just form elements.

Properties:

1. DOMEventTarget relatedTarget

Form Events

Event names:

1. onChange onInput onInvalid onSubmit

For more information about the onChange event, see [Forms](https://reactjs.org/docs/forms.html).

Mouse Events

Event names:

1. onClick onContextMenu onDoubleClick onDrag onDragEnd onDragEnter onDragExit
2. onDragLeave onDragOver onDragStart onDrop onMouseDown onMouseEnter onMouseLeave
3. onMouseMove onMouseOut onMouseOver onMouseUp

The onMouseEnter and onMouseLeave events propagate from the element being left to the one being entered instead of ordinary bubbling and do not have a capture phase.

Properties:

1. boolean altKey
2. number button
3. number buttons
4. number clientX
5. number clientY
6. boolean ctrlKey
7. boolean getModifierState(key)
8. boolean metaKey
9. number pageX
10. number pageY
11. DOMEventTarget relatedTarget
12. number screenX
13. number screenY
14. boolean shiftKey

Selection Events

Event names:

1. onSelect

Touch Events

Event names:

1. onTouchCancel onTouchEnd onTouchMove onTouchStart

Properties:

1. boolean altKey
2. DOMTouchList changedTouches
3. boolean ctrlKey
4. boolean getModifierState(key)
5. boolean metaKey
6. boolean shiftKey
7. DOMTouchList targetTouches
8. DOMTouchList touches

UI Events

Event names:

1. onScroll

Properties:

1. number detail
2. DOMAbstractView view

Wheel Events

Event names:

1. onWheel

Properties:

1. number deltaMode
2. number deltaX
3. number deltaY
4. number deltaZ

Media Events

Event names:

1. onAbort onCanPlay onCanPlayThrough onDurationChange onEmptied onEncrypted
2. onEnded onError onLoadedData onLoadedMetadata onLoadStart onPause onPlay
3. onPlaying onProgress onRateChange onSeeked onSeeking onStalled onSuspend
4. onTimeUpdate onVolumeChange onWaiting

Image Events

Event names:

1. onLoad onError

Animation Events

Event names:

1. onAnimationStart onAnimationEnd onAnimationIteration

Properties:

1. string animationName
2. string pseudoElement
3. float elapsedTime

Transition Events

Event names:

1. onTransitionEnd

Properties:

1. string propertyName
2. string pseudoElement
3. float elapsedTime

Other Events

Event names:

1. onToggle

---------------------------------------------<<<<Handling Events with Methods>>>>--------------------------------------

This key word is very important as it represent class, so properties and methods need to be used using this keyword.

Example:

<Person

          name={this.state.persons[0].name}

          age={this.state.persons[0].age}

        />

<button onClick={this.switchNamehandler}>Click!!!</button>

**Function variable:**

  //If we dont use function variable then in defination we will not able to use 'this'

  switchNamehandler = () => {

    this.setState({

      persons: [

        { name: "vatan", age: 20 },

        { name: "tanuja soni", age: 15 },

      ],

    });

  };