#_ Important [JavaScript Built-in Methods] {CheatSheet}

1. Array Methods

- **forEach()**: `array.forEach(element => console.log(element));`
- **map()**: `const squares = array.map(x => x * x);`
- **filter()**: `const evens = array.filter(x => x % 2 === 0);`
- **reduce()**: `const sum = array.reduce((acc, curr) => acc + curr, 0);`
- **find()**: `const found = array.find(x => x > 10);`
- **indexOf()**: `const index = array.indexOf(item);`
- **push()**: `array.push(item);`
- **pop()**: `const lastItem = array.pop();`
- **shift()**: `const firstItem = array.shift();`
- **unshift()**: `array.unshift(item);`
- **splice()**: `array.splice(startIndex, deleteCount, item);`
- **slice()**: `const sliced = array.slice(startIndex, endIndex);`
- **concat()**: `const mergedArray = array1.concat(array2);`
- **join()**: `const str = array.join(', ');`
- **includes()**: `const hasItem = array.includes(item);`
- **some()**: `const hasNegative = array.some(x => x < 0);`
- **every()**: `const allPositive = array.every(x => x > 0);`
- **sort()**: `array.sort((a, b) => a - b);`
- **reverse()**: `array.reverse();`

2. String Methods

- **charAt()**: `const char = str.charAt(index);`
- **concat()**: `const combinedStr = str1.concat(str2);`
- **includes()**: `const hasSubstring = str.includes(substring);`
- **indexOf()**: `const index = str.indexOf(substring);`
- **lastIndexOf()**: `const lastIndex = str.lastIndexOf(substring);`
- **match()**: `const matches = str.match(regex);`
- **repeat()**: `const repeatedStr = str.repeat(times);`
- **replace()**: `const replacedStr = str.replace(regex, newSubstr);`
- **search()**: `const index = str.search(regex);`
- **slice()**: `const slicedStr = str.slice(startIndex, endIndex);`

- **split()**: `const tokens = str.split(delimiter);`
- **substring()**: `const substring = str.substring(startIndex, endIndex);`
- **toLowerCase()**: `const lowerStr = str.toLowerCase();`
- **toUpperCase()**: `const upperStr = str.toUpperCase();`
- **trim()**: `const trimmedStr = str.trim();`

3. Object Methods

- **Object.keys()**: `const keys = Object.keys(obj);`
- **Object.values()**: `const values = Object.values(obj);`
- **Object.entries()**: `const entries = Object.entries(obj);`
- **Object.assign()**: `const newObj = Object.assign({}, obj1, obj2);`
- **Object.freeze()**: `const frozenObj = Object.freeze(obj);`
- **Object.isFrozen()**: `const isFrozen = Object.isFrozen(obj);`
- **Object.seal()**: `const sealedObj = Object.seal(obj);`
- **Object.isSealed()**: `const isSealed = Object.isSealed(obj);`
- **Object.create()**: `const newObj = Object.create(prototypeObj);`
- **Object.hasOwnProperty()**: `const hasProp = obj.hasOwnProperty(prop);`

4. Number Methods

- **Number.parseInt()**: `const intVal = Number.parseInt(string, radix);`
- **Number.parseFloat()**: `const floatVal = Number.parseFloat(string);`
- **Number.isNaN()**: `const isNaN = Number.isNaN(value);`
- **Number.isFinite()**: `const isFinite = Number.isFinite(value);`
- **Number.isInteger()**: `const isInteger = Number.isInteger(value);`
- **Number.toFixed()**: `const fixedStr = num.toFixed(digits);`

5. Math Functions

- **Math.abs()**: `const absolute = Math.abs(num);`
- **Math.ceil()**: `const roundedUp = Math.ceil(num);`
- **Math.floor()**: `const roundedDown = Math.floor(num);`
- **Math.round()**: `const rounded = Math.round(num);`
- **Math.max()**: `const max = Math.max(num1, num2, ...);`
- **Math.min()**: `const min = Math.min(num1, num2, ...);`
- **Math.pow()**: `const power = Math.pow(base, exponent);`

- **Math.sqrt()**: `const sqrt = Math.sqrt(num);`
- **Math.random()**: `const randomNum = Math.random();`

6. Global Functions

- **parseInt()**: `const intVal = parseInt(string, radix);`
- **parseFloat()**: `const floatVal = parseFloat(string);`
- **isNaN()**: `const isNaN = isNaN(value);`
- **isFinite()**: `const isFinite = isFinite(value);`

7. Date Methods

- **Date.now()**: `const now = Date.now();`
- **Date.parse()**: `const timestamp = Date.parse(dateString);`
- **new Date()**: `const date = new Date(year, month, day, ...);`

8. Regular Expression Methods

- **RegExp.test()**: `const matchesPattern = regex.test(string);`
- **RegExp.exec()**: `const match = regex.exec(string);`

9. JSON Methods

- **JSON.stringify()**: `const jsonString = JSON.stringify(obj);`
- **JSON.parse()**: `const obj = JSON.parse(jsonString);`

10. Web APIs

- **setTimeout()**: `const timeoutId = setTimeout(() => { /* code */ }, delay);`
- **clearTimeout()**: `clearTimeout(timeoutId);`
- **setInterval()**: `const intervalId = setInterval(() => { /* code */ }, delay);`
- **clearInterval()**: `clearInterval(intervalId);`

11. Promises and Asynchronous Programming

- **Promise.then()**: `promise.then(value => { /* code */ });`

- **Promise.catch()**: `promise.catch(error => { /* code */ });`
- **Promise.finally()**: `promise.finally(() => { /* code */ });`
- **Promise.all()**: `Promise.all([promise1, promise2]).then(values => { /* code */ });`
- **Promise.race()**: `Promise.race([promise1, promise2]).then(value => { /* code */ });`
- **async function**: `async function fetchData() { const data = await fetch(url); }`

12. Console Methods

- **console.log()**: `console.log('message', obj);`
- **console.warn()**: `console.warn('warning message');`
- **console.error()**: `console.error('error message');`
- **console.info()**: `console.info('info message');`
- **console.clear()**: `console.clear();`

13. Miscellaneous Methods

- **alert()**: `alert('Alert message');`
- **confirm()**: `const confirmed = confirm('Are you sure?');`
- **prompt()**: `const userInput = prompt('Enter your name:', 'Default Name');`
- **encodeURIComponent()**: `const encoded = encodeURIComponent(uriComponent);`
- **decodeURIComponent()**: `const decoded = decodeURIComponent(encodedUriComponent);`

14. DOM Manipulation (Web-Specific)

- **document.getElementById()**: `const element = document.getElementById('id');`
- **document.querySelector()**: `const element = document.querySelector('.class');`
- **document.querySelectorAll()**: `const elements = document.querySelectorAll('div');`
- **document.createElement()**: `const div = document.createElement('div');`

- **element.appendChild()**: `parentElement.appendChild(childElement);`
- **element.innerHTML**: `element.innerHTML = '<span>New content</span>';`
- **element.addEventListener()**: `element.addEventListener('click', eventHandler);`
- **element.style**: `element.style.color = 'blue';`
- **element.setAttribute()**: `element.setAttribute('data-custom', 'value');`
- **element.removeAttribute()**: `element.removeAttribute('data-custom');`

15. Event Handling

- **element.onclick**: `element.onclick = function() { /* code */ };`
- **element.addEventListener()**: `element.addEventListener('click', eventHandler);`
- **element.removeEventListener()**: `element.removeEventListener('click', eventHandler);`
- **event.preventDefault()**: `element.addEventListener('click', event => event.preventDefault());`

16. BOM (Browser Object Model) Methods

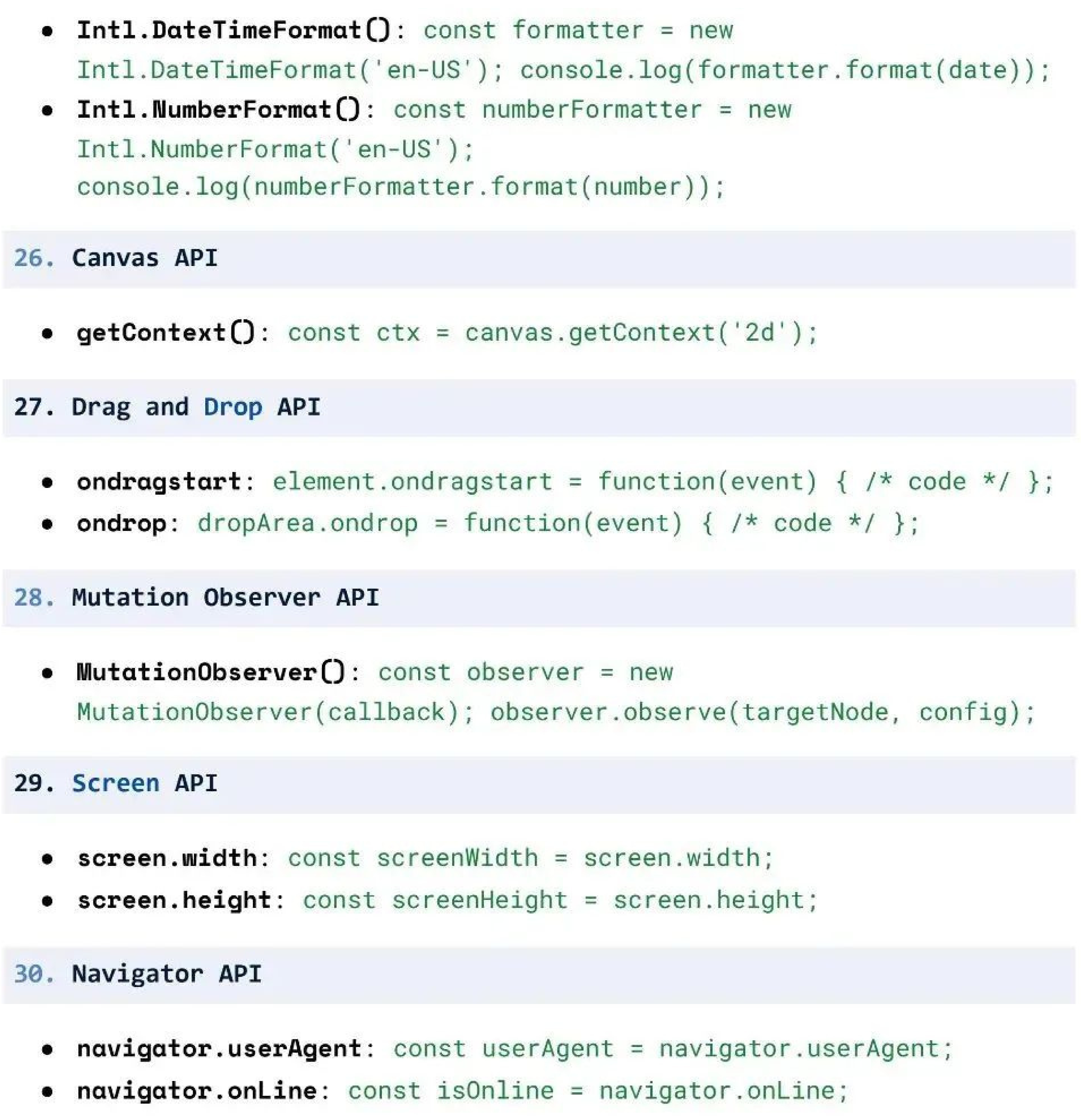
- **window.open()**: `const newWindow = window.open(url);`
- **window.close()**: `window.close();`
- **window.moveTo()**: `window.moveTo(x, y);`
- **window.resizeTo()**: `window.resizeTo(width, height);`

17. Web Storage API

- **localStorage.setItem()**: `localStorage.setItem('key', 'value');`
- **localStorage.getItem()**: `const value = localStorage.getItem('key');`
- **sessionStorage.setItem()**: `sessionStorage.setItem('key', 'value');`
- **sessionStorage.getItem()**: `const value = sessionStorage.getItem('key');`

18. Fetch API and XMLHttpRequest

- **fetch()**: `fetch(url).then(response => response.json());`
- **XMLHttpRequest()**: `const xhr = new XMLHttpRequest(); xhr.open('GET', url); xhr.send();`



- **`Intl.DateTimeFormat()`**: `const formatter = new Intl.DateTimeFormat('en-US'); console.log(formatter.format(date));`
- **`Intl.NumberFormat()`**: `const numberFormatter = new Intl.NumberFormat('en-US'); console.log(numberFormatter.format(number));`

26. Canvas API

- **`getContext()`**: `const ctx = canvas.getContext('2d');`

27. Drag and Drop API

- **`ondragstart`**: `element.ondragstart = function(event) { /* code */ };`
- **`ondrop`**: `dropArea.ondrop = function(event) { /* code */ };`

28. Mutation Observer API

- **`MutationObserver()`**: `const observer = new MutationObserver(callback); observer.observe(targetNode, config);`

29. Screen API

- **`screen.width`**: `const screenWidth = screen.width;`
- **`screen.height`**: `const screenHeight = screen.height;`

30. Navigator API

- **`navigator.userAgent`**: `const userAgent = navigator.userAgent;`
- **`navigator.onLine`**: `const isOnline = navigator.onLine;`

Tap here to get More Such notes on WhatsApp