

23 MODERN ES6 CODE SNIPPETS TO SOLVE PRACTICAL JS PROBLEMS



createDirIfNotExists.js

```
const fs = require('fs');
const createDirIfNotExists = dir => (!fs.existsSync(dir) ?
fs.mkdirSync(dir) : undefined);

// Example
createDirIfNotExists('test');
// creates the directory 'test', if it doesn't exist
```



```
isBrowserTabFocused.js

const isBrowserTabFocused = () => !document.hidden;

// Example
isBrowserTabFocused(); // true
```

```
copyToClipboard.js

const copyToClipboard = str => {
  const el = document.createElement('textarea');
  el.value = str;
  el.setAttribute('readonly', '');
  el.style.position = 'absolute';
  el.style.left = '-9999px';
  document.body.appendChild(el);
  const selected =
    document.getSelection().rangeCount > 0 ? document.getSelection().getRangeAt(0) :
false;
  el.select();
  document.execCommand('copy');
  document.body.removeChild(el);
  if (selected) {
    document.getSelection().removeAllRanges();
    document.getSelection().addRange(selected);
  }
};

// Example
copyToClipboard('Lorem ipsum'); // 'Lorem ipsum' copied to clipboard.
```

```
counter.js

const counter = (selector, start, end, step = 1, duration = 2000) => {
  let current = start,
      _step = (end - start) * step < 0 ? -step : step,
      timer = setInterval(() => {
        current += _step;
        document.querySelector(selector).innerHTML = current;
        if (current >= end) document.querySelector(selector).innerHTML = end;
        if (current >= end) clearInterval(timer);
      }, Math.abs(Math.floor(duration / (end - start))));
  return timer;
};

// Example
counter('#my-id', 1, 1000, 5, 2000);
// Creates a 2-second timer for the element with id="my-id"
```

```
httpPost.js

const httpPost = (url, data, callback, err = console.error) => {
  const request = new XMLHttpRequest();
  request.open('POST', url, true);
  request.setRequestHeader('Content-type', 'application/json; charset=utf-8');
  request.onload = () => callback(request.responseText);
  request.onerror = () => err(request);
  request.send(data);
};

const newPost = {
  userId: 1,
  id: 1337,
  title: 'Foo',
  body: 'bar bar bar'
};

const data = JSON.stringify(newPost);
httpPost(
  'https://jsonplaceholder.typicode.com/posts',
  data,
  console.log
);

// Logs: {"userId": 1, "id": 1337, "title": "Foo", "body": "bar bar bar"}
```

```
httpGet.js

const httpGet = (url, callback, err = console.error) => {
  const request = new XMLHttpRequest();
  request.open('GET', url, true);
  request.onload = () => callback(request.responseText);
  request.onerror = () => err(request);
  request.send();
};

httpGet(
  'https://jsonplaceholder.typicode.com/posts/1',
  console.log
);

// Logs: {"userId": 1, "id": 1, "title": "sample title", "body": "my text"}
```



getDaysDiffBetweenDates.js

```
const getDaysDiffBetweenDates = (dateInitial,  
dateFinal) =>  
  (dateFinal - dateInitial) / (1000 * 3600 * 24);  
  
// Example  
getDaysDiffBetweenDates(new Date('2017-12-13'), new  
Date('2017-12-22')); // 9
```



```
formatDuration.js

const formatDuration = ms => {
  if (ms < 0) ms = -ms;
  const time = {
    day: Math.floor(ms / 86400000),
    hour: Math.floor(ms / 3600000) % 24,
    minute: Math.floor(ms / 60000) % 60,
    second: Math.floor(ms / 1000) % 60,
    millisecond: Math.floor(ms) % 1000
  };
  return Object.entries(time)
    .filter(val => val[1] !== 0)
    .map(([key, val]) => `${val} ${key}${val !== 1 ? 's' : ''}`)
    .join(', ');
};

// Examples
formatDuration(1001); // '1 second, 1 millisecond'
formatDuration(34325055574); // '397 days, 6 hours, 44 minutes, 15 seconds, 574 milliseconds'
```



removeListener.js

```
const off = (el, evt, fn, opts = false) =>
  el.removeEventListener(evt, fn, opts);

const fn = () => console.log('!');
document.body.addEventListener('click', fn);
off(document.body, 'click', fn);
// no longer logs '!' upon clicking on the page
```

```
triggerEvent.js

const triggerEvent = (el, eventType, detail) =>
  el.dispatchEvent(new CustomEvent(eventType, { detail }));

// Examples
triggerEvent(document.getElementById('myId'), 'click');
triggerEvent(document.getElementById('myId'), 'click', { username:
'bob' }); 'selector.to.val', 'target[0]', 'target[2].a');

// ['val to select', 1, 'test']
```

```
getProperties.js

const get = (from, ...selectors) =>
  [...selectors].map(s =>
    s
      .replace(/\([([^\[\]]*)\]/g, '.$1.')
      .split('.')
      .filter(t => t !== '')
      .reduce((prev, cur) => prev && prev[cur], from)
  );
const obj = { selector: { to: { val: 'val to select' } }, target: [1, 2, { a: 'test' }] };

// Example
get(obj, 'selector.to.val', 'target[0]', 'target[2].a'); // ['val to select', 1, 'test']
```



formToObject.js

```
const formToObject = form =>
  Array.from(new FormData(form)).reduce(
    (acc, [key, value]) => ({
      ...acc,
      [key]: value
    }),
    {}
  );

// Example
formToObject(document.querySelector('#form'));
// { email: 'test@email.com', name: 'Test Name' }
```

```
getURLParameters.js

const getURLParameters = url =>
  (url.match(/(?:^?=&+)(=([^&]*))/g) || []).reduce(
    (a, v) => ((a[v.slice(0, v.indexOf('='))]) =
v.slice(v.indexOf('=') + 1)), a),
    {}
  );

// Examples
getURLParameters('http://url.com/page?n=Adam&s=Smith');
// {n: 'Adam', s: 'Smith'}
getURLParameters('google.com'); // {}
```



```
detectDeviceType.js

const detectDeviceType = () =>
  /Android|webOS|iPhone|iPad|iPod|BlackBerry|IEMobile|Opera
Mini/i.test(navigator.userAgent)
    ? 'Mobile'
    : 'Desktop';

// Example
detectDeviceType(); // "Mobile" or "Desktop"
```

```
getImages.js

const getImages = (el, includeDuplicates = false) => {
  const images = [...el.getElementsByTagName('img')].map(img =>
img.getAttribute('src'));
  return includeDuplicates ? images : [...new Set(images)];
};

// Examples
getImages(document, true);
// ['image1.jpg', 'image2.png', 'image1.png', '...']
getImages(document, false);
// ['image1.jpg', 'image2.png', '...']
```




currentURL.js

```
const currentURL = () => window.location.href;
```

```
// Example
```

```
currentURL(); // 'https://google.com'
```

```
elementsVisibleInViewport.js

const elementIsVisibleInViewport = (el, partiallyVisible = false) => {
  const { top, left, bottom, right } = el.getBoundingClientRect();
  const { innerHeight, innerWidth } = window;
  return partiallyVisible
    ? ((top > 0 && top < innerHeight) || (bottom > 0 && bottom < innerHeight)) &&
      ((left > 0 && left < innerWidth) || (right > 0 && right < innerWidth))
    : top >= 0 && left >= 0 && bottom <= innerHeight && right <= innerWidth;
};

// Examples
elementIsVisibleInViewport(el); // (not fully visible)
elementIsVisibleInViewport(el, true); // (partially visible)
```



elementContains.js

```
const elementContains = (parent, child) => parent !== child &&  
parent.contains(child);
```

```
// Examples
```

```
elementContains(document.querySelector('head'),  
document.querySelector('title'));
```

```
// true
```

```
elementContains(document.querySelector('body'),  
document.querySelector('body')); // false
```

```
scrollTop();
```



```
scrollToTop.js

const scrollToTop = () => {
  const c = document.documentElement.scrollTop || document.body.scrollTop;
  if (c > 0) {
    window.requestAnimationFrame(scrollToTop);
    window.scrollTo(0, c - c / 8);
  }
};

// Example
scrollToTop();
```



getScrollPosition.js

```
const getScrollPosition = (el = window) => ({  
  x: el.pageXOffset !== undefined ? el.pageXOffset : el.scrollLeft,  
  y: el.pageYOffset !== undefined ? el.pageYOffset : el.scrollTop  
});
```

```
// Example
```

```
getScrollPosition(); // {x: 0, y: 200}
```



toggleClass.js

```
const toggleClass = (el, className) =>
  el.classList.toggle(className);

// Example
toggleClass(document.querySelector('p.special'),
  'special');
// The paragraph will not have the 'special' class
anymore
```



hasClass.js

```
const hasClass = (el, className) =>
  el.classList.contains(className);

// Example
hasClass(document.querySelector('p.special'),
  'special'); // true
```



hide.js

```
const hide = (...el) => [...el].forEach(e =>  
(e.style.display = 'none'));
```

```
// Example
```

```
hide(document.querySelectorAll('img'));
```

```
// Hides all <img> elements on the page
```


THANK
YOU

FOLLOW FOR MORE SUCH
INTERESTING POST



Linked in

<http://www.linkedin.com/in/depaksharma/>