



BABY WOLF CODES

📌 javascript

# Import, Export

- CHEATSHEET



# Named exports.

Declaring and exporting at the same time



util.js

```
export const VERSION = '1.1';  
  
export function parse(user) {  
  // ...  
}
```



main.js

```
import {VERSION, parse} from './util.js';
```

# Named exports.

Declaring first and exporting later



util.js

```
const VERSION = '1.1';

function parseUser(user) {
  // ...
}

export {
  VERSION,
  parseUser as parse,
}
```

Alias



main.js

```
import {VERSION, parse} from './util.js';
```



# Importing all.



util.js

```
export function parse(user) {  
  // ...  
}
```



main.js

```
import * as utils from './util.js'  
  
utils.parse(someData);
```

# Importing w/ alias.



util.js

```
export function parse(user) {  
  // ...  
}
```



main.js

```
import {parse as parseUser}  
  from './util.js'  
  
parseUser(someData);
```

# Default exports.

Not recommended, avoid if possible



util.js

```
export default class {  
  parse(user) {  
    // ...  
  }  
}
```



main.js

```
import UserUtils from './util.js';  
  
new UserUtils().parse(someData);
```

Notice that `{ }` is not used during import



# Mixed style.

Both default and named exports



util.js

```
export default function() { /* ... */ }  
export function foo() { /* ... */ }
```



main.js

```
import bar, { foo } from './util.js';  
  
foo();  
bar(); // Default function
```



BABY WOLF CODES

# Thanks for reading!

💬 SHARE YOUR THOUGHTS



If you find my content helpful,  
consider donating by clicking on  
this highlight in my profile



Support