

Cheat Sheet for comprehensive Babel

Installation and Setup

Installing Babel

- Global Installation:

```
npm install -g @babel/core @babel/cli
```

- Local Installation:

```
npm install --save-dev @babel/core @babel/cli
```

Configuration Files

- babel.config.json:

```
{
  "presets": ["@babel/preset-env"]
}
```

- .babelrc:

```
{
  "presets": ["@babel/preset-env"]
}
```

Core Concepts

Presets

- @babel/preset-env:

- Transpiles ES6+ to ES5 based on target environments.
- Example:

```
{
  "presets": [
    "@babel/preset-env", {
      "targets": "> 0.25%, not dead"
    }
  ]
}
```

```
  ]  
}
```

- **@babel/preset-react:**

- Transpiles JSX to JavaScript.
- Example:

```
{  
  "presets": ["@babel/preset-react"]  
}
```

Plugins

- **@babel/plugin-transform-arrow-functions:**

- Transforms arrow functions to ES5 compatible functions.
- Example:

```
{  
  "plugins": ["@babel/plugin-transform-arrow-functions"]  
}
```

- **@babel/plugin-proposal-class-properties:**

- Transforms class properties.
- Example:

```
{  
  "plugins": ["@babel/plugin-proposal-class-properties"]  
}
```

CLI Commands

Basic Commands

- **Transpile a Single File:**

```
npx babel input.js --out-file output.js
```

- **Transpile and Watch:**

```
npx babel input.js --watch --out-file output.js
```

- **Transpile Directory:**

```
npx babel src --out-dir lib
```

Advanced Commands

- **Ignore Files:**

```
npx babel src --out-dir lib --ignore "src/**/*.spec.js"
```

- **Copy Files:**

```
npx babel src --out-dir lib --copy-files
```

Configuration Options

Targeting Environments

- **Browserslist:**

- Define in ``.browserslistrc`` or ``.package.json``.
- Example:

```
{
  "browserslist": [
    "> 1%",
    "last 2 versions"
  ]
}
```

Polyfills

- **@babel/polyfill:**

- Provides polyfills for ES6+ features.
- Example:

```
npm install --save @babel/polyfill
```

- Usage:

```
import "@babel/polyfill";
```

Debugging and Troubleshooting

Verbose Output

- **Enable Verbose Mode:**

```
npm run babel src --out-dir lib --verbose
```

Error Handling

- **Common Errors:**

- **Syntax Error:** Check for unsupported syntax in your code.
- **Plugin/Preset Not Found:** Ensure all plugins/presets are installed.

Advanced Features

Custom Plugins

- **Creating a Plugin:**

- Example:

```
module.exports = function(api) {  
  api.cache(true);  
  return {  
    visitor: {  
      Identifier(path) {  
        if (path.node.name === "foo") {  
          path.node.name = "bar";  
        }  
      }  
    }  
  };  
};
```

Custom Presets

- **Creating a Preset:**

- Example:

```
module.exports = function() {
  return {
    presets: [
      require("@babel/preset-env")
    ],
    plugins: [
      require("@babel/plugin-proposal-class-properties")
    ]
  };
};
```

Integration with Build Tools

Webpack

- babel-loader:

- Install:

```
npm install --save-dev babel-loader
```

- Configuration:

```
module.exports = {
  module: {
    rules: [
      {
        test: /\.js$/,
        exclude: /node_modules/,
        use: {
          loader: "babel-loader"
        }
      }
    ]
  }
};
```

Gulp

- gulp-babel:

- Install:

```
npm install --save-dev gulp-babel
```

- Configuration:

```
const gulp = require("gulp");
const babel = require("gulp-babel");

gulp.task("babel", () =>
  gulp.src("src/*.js")
    .pipe(babel())
    .pipe(gulp.dest("dist"))
);
```

Tips and Best Practices

Code Splitting

- Dynamic Imports:

- Use dynamic imports for better performance.
- Example:

```
import("./module").then(module => {
  module.default();
});
```

Minification

- Minify Output:

- Use `@babel/preset-env` with `useBuiltIns: "usage"` for minimal polyfills.
- Example:

```
{
  "presets": [
    ["@babel/preset-env", {
      "useBuiltIns": "usage"
    }]
  ]
}
```

Continuous Integration

- CI/CD Integration:

- Ensure Babel is part of your CI/CD pipeline.
- Example:

```
script:
  - npm install
  - npx babel src --out-dir lib
```

Additional Resources

Official Documentation

- [Babel Documentation](<https://babeljs.io/docs/>)

Community and Support

- [Babel GitHub Repository](<https://github.com/babel/babel>)
- [Babel Forum](<https://babeljs.io/forum/>)

Summary

- **Core Concepts:** Presets, Plugins, and Configuration.
- **CLI Commands:** Basic and advanced usage.
- **Integration:** Webpack, Gulp, and CI/CD.
- **Best Practices:** Code splitting, minification, and continuous integration.

This cheat sheet provides a comprehensive overview of Babel, covering installation, configuration, CLI commands, advanced features, and best practices. Use it as a quick reference for all your Babel needs.

By Ahmed Baheeg Khorshid

ver 1.0