JavaScript Events Cheat Sheet

Hubino

Event Listener Registration

You can register event listener to handle event triggered by HTML elements. This is typically done using the addEventListener method.

index.js

```
const button = document.getElementById(’myButton’);

button.addEventListener(’click’, function() {
  // your event handling code here
});
```

Hubino

Event Types

There are various types of events, such as click, mouseover, keydown, and submit, to name a few. Choose the appropriate event type based on the user interaction you want to capture.

index.js

```
element.addEventListener('mouseover', function(){
  // Mouseover event handling
});
```

Hubino

Event Object

Event listeners receive an event object as an argument. This object contains information about the event, such as the target element and event type.

index.js

```
button.addEventListener('click', function(event) {
console.log('Button clicked: ", event.target);
});
```

Hubino

Event Bubbling

Event propagate upwards through the DOM tree by default. You can stop this propagation using the stopPropagation methos

index.js

```
parentElement.addEventListener(’click’, function() {
  //This event fires for child and parent elements
});

childElement.addEventListener(’click’, function(event) {
event.stopPropagation();
});
```

Hubino

Prevent Default

You can provent the default action associated with an event (e.g, form submission or link navigation) using the preventDefault method.

index.js

```
anchorElement.addEventListener('click', function
(event) {
event.preventDefault();
});
```

Hubino

Removing Event Listeners

You can remove event listeners usinf the removeEventListener method. Ensure the function reference matches the one you used to add the listener.

index.js

```
function eventHandler() {
//Event handling code
}

element.addEventListener('click', eventHandler);
element.removeEventListener('click', eventHandler);
```

Hubino

Event Delegation

Event delegation is a technique where you attch a single event listener to a parent element to handle events for its children. This is efficient for dynamically generated content.

index.js

```
parentElement.addEventListener('click',
function(event) {
if(event.target.matches('button')) {
}
});
```

Hubino

Keyboard Events

You can capture keyboard events, such as keydown, keyup, and keypress, to respond to user imput from the keyboard.

index.js

```
document.addEventListener('keydown',
function(event) {
if(event.key =='Enter') {
//Handle enter key press
}
});
```

Hubino