**XD Testing Tools**

**Connecting Devices**

**Connecting real devices:** Click on the “QR code”-button on the top of the page and then scan the QR code with the device you want to connect.

**Auto-connecting:** Toggle the “Auto-connect” checkbox of a device in the “Sessions”-area to automatically connect all newly created devices to the device.

**Manually connecting:** Open the settings menu of the device and choose the device you want to connect to from the dropdown list.

**Emulating Devices**

**Adding a new device:** Click the “+”-button in the menu on top. Either select a predefined device by typing in the “Predefined devices” input field or specify a custom device.

**Moving the device:** Start dragging around the border or title of the device to change the position.

**Settings of the device:** Click on the settings button on the top right of the device to show the settings menu. In the settings you can change the scaling of the device (does not change resolution), switch the orientation or refresh the device. You can also connect the device to another device.

**Changing the resolution:** You can change the resolution by moving the cursor to the bottom right of the iframe of a device.

**XD Testing Tools**

**XD Testing Tools**

**Function Debugging**

**Starting function debugging:** After adding all emulated devices that you want to debug, open the Chrome Developer tools (if they are already open, close them and re-open them). You can then add functions to debug next to the JavaScript console.

**Debug functions:** If a function is added to function debugging and then called from any active emulated device, execution will be paused, similar to setting a breakpoint in the first line of the function. You can also take a look at the function by clicking on the “Inspect”-button next to its name.

**XD Testing Tools**

**Shared JavaScript Console**

**Activating/deactivating devices:** Above the console, you see the names of all devices. By clicking on a device you can activate or deactivate it. Each active device has its own unique color which is used to color-code the console outputs from the devices.

**Sending commands:** Type a command into the input field at the bottom of the console and it will be executed on all active devices.

**Aggregating console outputs:** The console aggregates the console outputs from all active devices. This includes functions like console.log, console.warn, … It also displays the return values of all JavaScript commands that were sent to the devices. It also forwards all JavaScript errors that occur during execution.

**XD Testing Tools**

**Shared CSS Editor**

**Adding rules:** To add a rule, first type the selector in the editor and then the properties and values that you want to add.

**Changing rules:** Selectors/properties/values can be edited by clicking on them and then changing it.

**Enabling/disabling rules:** Rules can be disabled/enabled by toggling the checkbox next to the rule.

**Removing rules:** Click on the remove button on the top right of the rule.

**Chrome DevTools**

**JavaScript Console**

**Opening the console:** Open Chrome DevTools and then click on the tab “Console”.

**Calling functions:** You can call functions from the console. The return values will be displayed in the console.

**Inspecting variables:** You can access variables by typing them in the console.

**Console output:** The console will display all JS errors, console logs, …

**JavaScript Debugging**

**Starting debugging:** Open Chrome DevTools and click on the “Sources” tab. You can see a list of domains and associated files on the left. Navigate to the file that you want to debug.

**Setting breakpoints:** If you opened a JS file, you can click on the line number to set a breakpoint at this line. If the breakpoint is reached, you can do various things such as inspecting variables. You can also step through instructions one by one.

**Blackboxing scripts:** You can right click on a JS file and select the option “Blackbox script”. The script will then be skipped when stepping through an execution. This is especially useful for libraries.

**Chrome DevTools**

**Chrome DevTools**

**HTML and CSS Inspection**

**Starting inspection:** Open Chrome DevTools and click on the “Elements” tab or right-click an element on the webpage and choose “Inspect element”.

**Editing HTML:** Right-click on the element and you will see various options for editing, e.g. adding attributes or editing as HTML.

**Editing CSS:** When you select an HTML element, you can see (and edit) its CSS to the right. You can also adding completely new rules by clicking on the “plus” button on the top right. You can enable/disable rules by hovering over them and toggling the checkbox to the left.

**Chrome DevTools**

**Device Mode**

**Opening device mode:** Open Chrome DevTools and click on the phone-icon on the top left. You need to refresh the page to display properly after changing the device.

**Emulating a device:** You can select the device you want to emulate on the top left. You can then edit the resolution by changing the numbers on the top left or by clicking on the right/bottom/bottom-right of the device and dragging.

**Record/Replay**

**Recording interactions:** On the right side of the tools, you see the record/replay area. Click on the record button of a specific device, then perform the desired interactions and click the same button again.

**Saving event sequences:** You can save an event sequence by clicking on the save button at the top right of the event sequence, then typing a name and hitting enter.

**Appending event sequences:** To add a saved event sequence to a device, select it from the dropdown menu at the top of the device’s section in the record/replay area.

**Adjusting event sequences:** Once you have recorded an event sequence, you can drag and drop it to a later point in the timeline or to another device.

**Adding breakpoints:** To add a breakpoint, click at the desired location of the breakpoint to the left of the timeline labels between the two thin vertical lines. To remove the breakpoint again, start dragging it and drop it into the icon that will be highlighted on top. When a breakpoint is reached during replay, the replaying of interactions will pause until you click on the play button on top of the breakpoints.

**Replaying event sequences:** To replay an event sequence on a specific device, click that device’s replay button. To replay the sequences on all devices, click the play button on top of the page.