# User Study

## Type of Study

**Within-subjects:**

* Each participant solves half the task with the tool and half the tasks without the tool
* Disadvantages:
  + Carryover effects

**Between-subjects:**

* Half the participants solve all the tasks with the tool and half the participants solve all tasks without the tool
* Advantages:
  + Possibly shorter time for the study as we need less tasks per participant (randomly choose one application for each participant)
* Disadvantages:
  + More participants required
  + No direct comparison between tool/no tool in questionnaire possible

## Setup

* One desktop device with keyboard, mouse and one screen
* One tablet, one smartphone
* Without the tool participants have access to:
  + Chrome Developer Tools
  + Chrome Device Mode
  + Remote debugging of real devices (devices are already connected to Chrome before the study begins)
* With the tool, participants have access to:
  + Tool
  + Chrome Developer Tools
  + Chrome Device Mode
  + Remote debugging of real devices
* Cheat sheet about tools, dev tools, …

## Introduction

Demonstration of the cross-device applications used in the study

Tool:

* Show how to create emulated devices and connect devices
* Show how to connect real devices
* Show how the JS Console and function debugging works
* Mention that Dev Tools need to be opened after devices have loaded for function debugging
* Show how record/replay works

General:

* Explain Chrome Device Mode
* Explain Chrome Developer Tools: console, HTML/JS/CSS inspection/debugging
* Explain how to remotely debug a real device
* Show users how to make multiple profiles in Chrome, possibly already provide a few profiles

## Evaluation

* Measure time/success rate for the different tasks with/without tools
* Logging actions of participants:
  + How often do participants use stuff like device mode when they have access to the tool compared to when they do not have access to the tool
* Screen recording to log features that the participants use
* Keep count of how often a user uses a feature in case screen recording goes wrong
* Questionnaire about the tasks
  + How easy/difficult was it to fix a bug/implement a new feature with/without the tool?
  + How efficient was it to fix a bug/implement a new feature with/without the tool?
* General questionnaire
  + Age
  + Gender
  + Previous experience with developing web application:
    - Responsive design
    - Cross-Device
    - How many years of experience? How many hours on average per week?
  + Experience with JavaScript in particular:
    - How many years?
    - Complexity of applications
    - How do you rate your skills from basic to expert?
  + Experience with Chrome Developer Tools
    - Also for particular features
  + Experience with Chrome Device Mode
* Concluding questionnaire
  + Suggestions for additional features (mention the features that we already have but that were deactivated for the study)
  + Suggestions for improvements
  + Easier/more efficient with or without?
  + Did you prefer debugging/implementing with or without the tool?
  + Was it easy to learn how to use the tool?
  + I thought the system was easy to use
  + The tool would be useful for debugging/developing cross-device applications
  + I would use the tool for debugging/developing cross-device applications
  + I felt very confident using the system
  + I thought there was too much inconsistency in the system
  + I found the various functions in the system were well integrated
  + The tool was unnecessarily complex
  + Which feature did you find most useful?
  + Which feature did you not find useful at all?
  + General comments

## Tasks

* Two applications
* Choose between jQuery/no jQuery?
* For each application
  + Fix a bug
  + Implement a new feature
  + Describe all steps that need to work correctly to count the feature as completed
* Fixing a bug:
  + Describe the steps needed to reach the bug in the application
  + The bug should probably cause some exception in JavaScript so the participants have a starting point
  + Screenshot of the bug (if useful)
  + Moving all functions that are relevant for fixing the bug to a separate file?
* Implementing a new feature:
  + Describe exactly how the feature should work, maybe provide a mockup
  + Skeleton of the function + necessary interface elements are given
* Users get a maximum time per task, if they cannot solve it successfully during that time, the task is aborted
  + Fixing a bug: 15-20 minutes
  + Implementing a new feature: 30 minutes
  + Total: 90-110 minutes + explanations + questionnaires 🡪 about 2 hours maximum
* Should we check if the participant has indeed solved the task correctly and let him try again if not?
* Should we intercept when participants are stuck for a long time or do not make any progress at all at the beginning?
  + Time to complete the task could not be counted towards the result, but the results from the questionnaire could be more useful
  + If the user does not do anything for the entire time of the task, the results from the questionnaire would also not be valid

## Applications

* XDCinema
  + Possible additional feature:
    - In the movie view, display the price of the movie in each cinema and highlight the correct price when a cinema is selected
    - In the “extra” role, display additional information about the cinema when a cinema is selected, e.g. parking, public transport, nearby restaurants, picture of the cinema, …
    - ~~Add button to show comments about a movie~~ 🡪 Requires only one device, not useful
    - Display list of nearby restaurants with name + rating when choosing a movie in extra role
  + Possible bug fixes
    - Bug: Some data is not properly synchronized
    - …
* XDYouTube
  + Possible additional features:
    - Add a play/pause button to landscape view
    - Go to first page of videos
    - Show multiple videos in parallel: All but one device have the video role and always take the first video from the queue
  + Possible bug fixes
    - Bug: If a video has finished, always take the next video and update the current video, even if the queue is empty
    - Bug: if the user clicks on the next page/previous page button, always show the first page
    - Bug: error when the user clicks on the previous page button while on the first page
    - Bug: queue not properly synchronized
* Gallery
  + Possible additional features
    - Add title/category to images
    - Mark images as favorite and browse favorite images
    - Go to next/previous picture on devices that only show the picture
  + Possible bug fixes
    - …
* Maps
  + Possible additional features
    - …
  + Possible bug fixes
    - …

## Variables

* Application:
  + Application 1
  + Application 2
* Task:
  + Fixing a bug
  + Implementing a feature
* Setup:
  + With tool
  + Without tool

## Features that should be deactivated

Some features should be deactivated so the participants are not overwhelmed:

* CSS Editor (not essential for the type of tasks in the study)
* Loading URLs (the application is given by the task anyway)
* Settings (settings are given by the study setup)
* Maybe inspect HTML (not essential for the type of tasks in the study)
* Scoping (unless strictly required for the tasks)
* Record/replay?

## Task Completion Times

Questionnaires: about 10 minutes

XDCinema, displaying and highlighting prices: about 8 minutes