

Client-Side Web Development

Class 3.1

Today's Topics

- CSS Grid
- **Exercise:** Grid Garden

Announcements

In-Class Labs

Any Questions?

CSS Grid

CSS Grid is a powerful layout system which allows for the control and placement of items into columns and rows

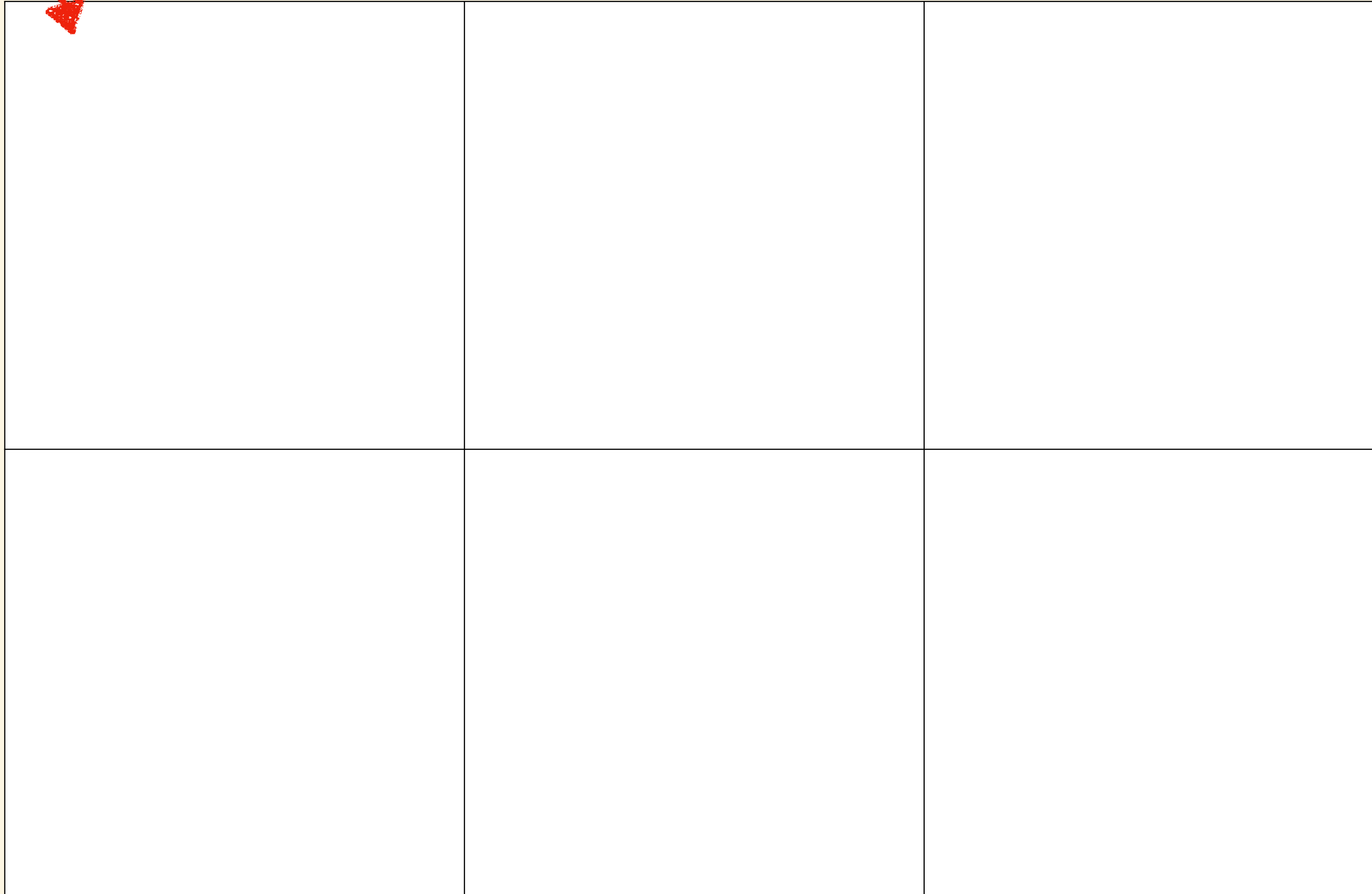
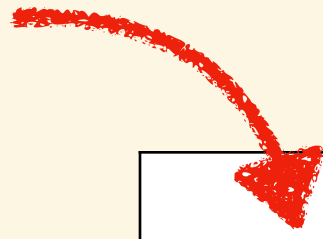
When working with *CSS Grid*, CSS rules will be applied to both the parent (Grid Container) and the *DIRECT* children (Grid Items)

CSS Grid Terminology

CSS Grid Terminology

- **Grid Container:** The element that `display: grid` is applied
- **Grid Item:** The direct child of a grid container
- **Grid Line:** The dividing lines that make up the grid structure, which run both vertically (columns) and horizontally (rows)

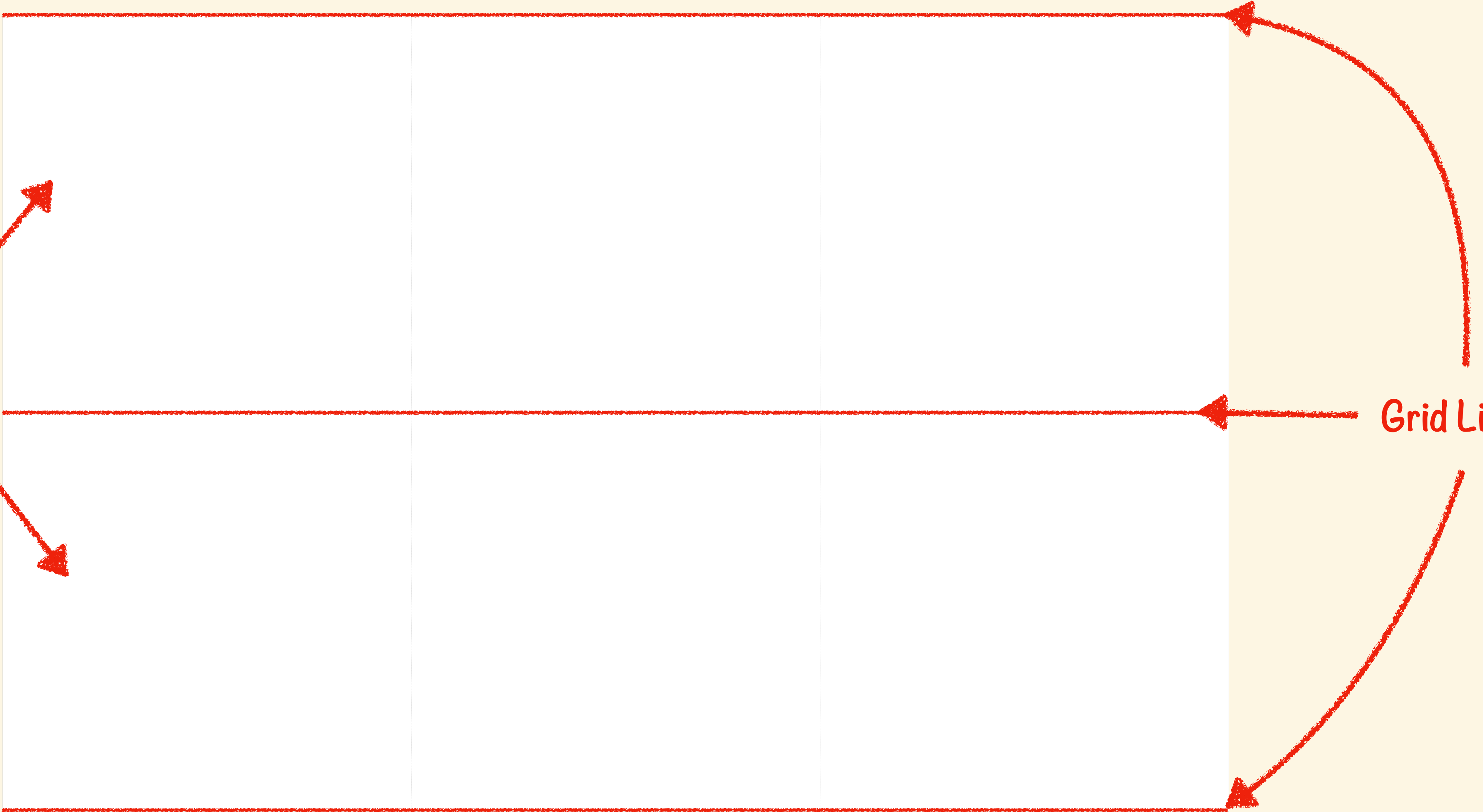
A CSS Grid



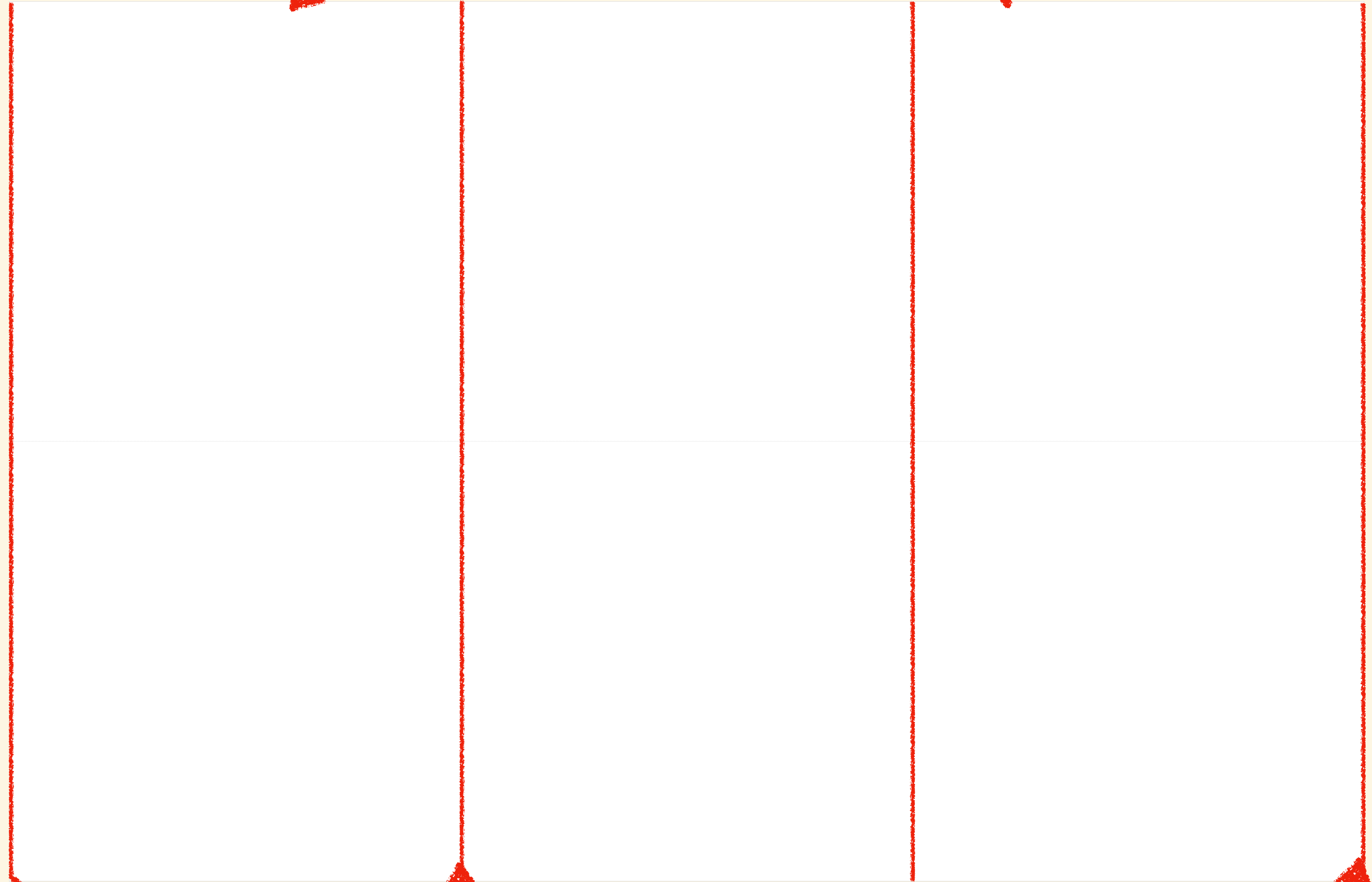
Grid Rows



Grid Lines



Grid Columns



Grid Lines



	1	2	3	4
1				
2				
3				

CSS Grid Terminology

- **Grid Container:** The element that `display: grid` is applied
- **Grid Item:** The direct child of a grid container
- **Grid Line:** The dividing lines that make up the grid structure, which run both vertically (columns) and horizontally (rows)
- **Grid Track:** The space between two grid lines (aka a single column or a single row)
- **Grid Cell:** A single unit of the grid
- **Explicit Grid:** The grid column and rows defined in CSS Rules.
- **Implicit Grid:** The grid column and rows auto-generated by the browser, but not defined

Defining a Grid

Use the `display: grid` to create a **Grid container**

Use the `grid-template-columns` and `grid-template-rows` to create an *explicit* grid

These properties take a list of sizes, with each item in the list becoming an explicit row or column

The `repeat()` function can be used to create multiple columns or rows of the same size. The first value of function is the number of times to repeat, the second the size.

The `grid-gap` property can be used to add space between grid cells

Examples

Grid Item Placements

Use the `grid-column-start` or `grid-row-start` properties to specify on which *grid line* a *grid item* should start

Use the `grid-column-end` or `grid-row-end` properties to specify on which *grid line* a *grid item* should end

If only a start or an end is provided the *grid item* will span **1** grid column or row

The shortcuts `grid-column` and `grid-row` will take both the start and end separate by a `/`

The **span** keyword followed by an integer can be used instead of a **grid line** number to indicate how many rows or columns to span

Examples

Using Grid Areas

The `grid-template-areas` property is used to specify areas of a grid using a grid area name

The . can be used to represent a part of the grid where there is no grid areas

Use the `grid-area` property on a grid item to place that item in the grid area

Examples

Auto Fit and Fill

The `auto-fill()` function fills the row with as many columns as it can fit, creating implicit columns whenever a new column can fit

The `auto-fit()` function fits the currently available columns into the space by expanding them to take up the available space

The `minmax()` function define a range of sizes
of a column or row

Examples

Alignment

Use `justify-items` and `align-items` properties to align the content horizontally and vertically, respectively, inside each grid cell

Use `justify-content` and `align-content` to align the grid cells horizontally and vertically, respectively, inside the grid

Use `justify-self` and `align-self` on a grid item to align the content horizontally and vertically, respectively, inside each grid cell

Nested Grids

Exercise: Grid Garden

For next class...

- **Review:** Keeping Things Accessible
- **Lab:** Stay on the Grid