timeliney

Pedro Alves

Table of contents

```
#timeline(body,
  spacing: 5pt,
  show-grid: false,
  grid-style: (
  stroke: (
    dash: "dashed",
    thickness: 0.5pt,
    paint: luma(66.67%),
  ),
),
  task-vline: true,
  line-style: (stroke: 3pt),
  milestone-overhang: 5pt,
  milestone-layout: "in-place",
  box-milestones: true,
  milestone-line-style: (),
  offset: 0)
```

Argument

`spacing: 5pt` length

Spacing between lines

Argument

`show-grid: true` boolean

Show a grid behind the timeline

Argument

```
grid-style: (
  stroke: (
    dash: "dashed",
    thickness: 0.5pt,
    paint: luma(66.67%),
  ),
)
```

dictionary

The style to use for the grid (has no effect if `show-grid` is false)

Argument

`task-vline: true` boolean

Show a vertical line next to the task names

Argument

`line-style: (stroke: 3pt)` dictionary

The style to use for the lines in the timelines

Argument

`milestone-overhang: 5pt` length

How far the milestone lines should extend past the end of the timeline (only has an effect if `milestone-layout` is `in-place`)

Argument

`milestone-layout: "in-place"` `string`

How to lay out the milestone lines. Can be `in-place` or `aligned`.

`in-place` displays the milestones directly below the timeline, and tries to lay them out as well as possible to avoid colisions.

`aligned` displays the milestones in a separate box, aligned with the task titles.

Argument

`box-milestones: true` `boolean`

Whether to draw a box around the milestones (only has an effect if `milestone-layout` is `aligned`)

Argument

`milestone-line-style: ()` `dictionary`

The style to use for the milestone lines

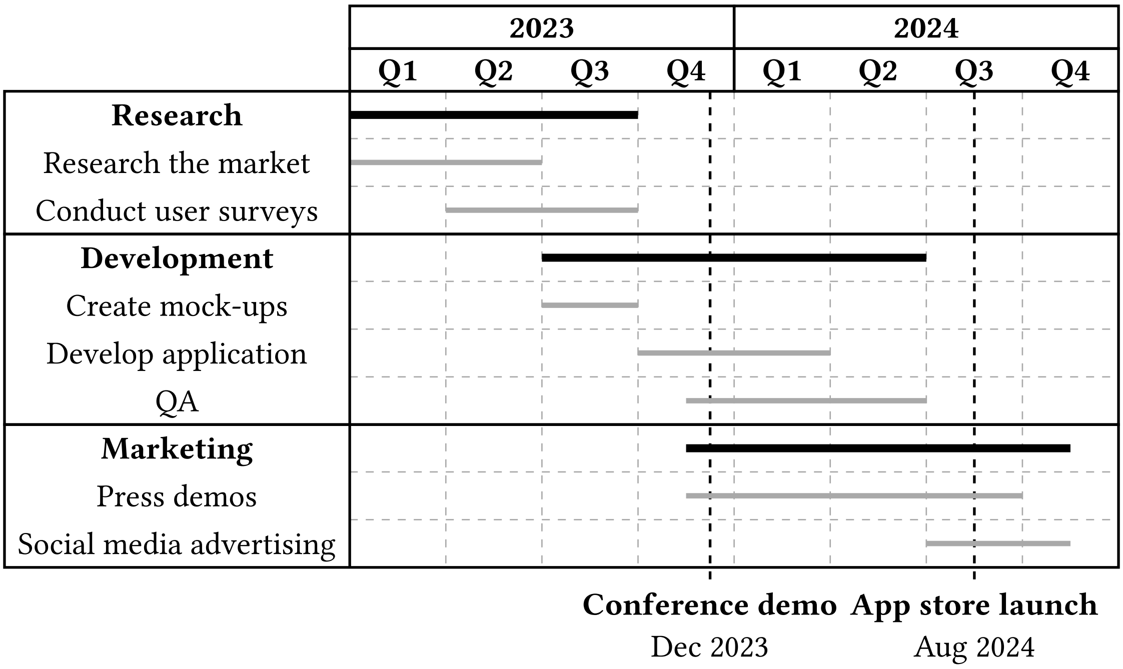
Argument

`offset: 0` `float`

Offset to be automatically added to all the timespans

```
#timeline(
show-grid: true,
{
  headerline(group(([*2023*], 4)), group(([*2024*], 4)))
  headerline(
    group(..range(4).map(n => strong("Q" + str(n + 1)))),
    group(..range(4).map(n => strong("Q" + str(n + 1)))),
  )

  taskgroup(title: [*Research*], {
    task("Research the market", (0, 2), style: (stroke: 2pt +
gray))
    task("Conduct user surveys", (1, 3), style: (stroke: 2pt +
gray))
  })

  taskgroup(title: [*Development*], {
    task("Create mock-ups", (2, 3), style: (stroke: 2pt + gray))
    task("Develop application", (3, 5), style: (stroke: 2pt +
gray))
    task("QA", (3.5, 6), style: (stroke: 2pt + gray))
  })
```

```
    taskgroup(title: [*Marketing*], {
      task("Press demos", (3.5, 7), style: (stroke: 2pt + gray))
      task("Social media advertising", (6, 7.5), style: (stroke: 2pt + gray))
    })

    milestone(
      at: 3.75,
      style: (stroke: (dash: "dashed")),
      align(center, [
        *Conference demo*\
        Dec 2023
      ])
    )

    milestone(
      at: 6.5,
      style: (stroke: (dash: "dashed")),
      align(center, [
        *App store launch*\
        Aug 2024
      ])
    )
  }
)
```



#headerline(..titles)

Argument

`..titles` `array`

The titles to display in the header line.

Can be specified in several different formats:

```
// One column per title
#headerline("Title 1", "Title 2", "Title 3")
```

```
(
  (
    type: "header",
    headers: (
      (
        titles: (("Title 1", 1), ("Title 2", 1), ("Title 3", 1)),
      ),
    ),
    total: 3,
  ),
)
```

```
// Each title occupies 2 columns
#headerline(("Title 1", 2), ("Title 2", 2))
```

```
(
  (
    type: "header",
    headers: ((titles: (("Title 1", 2), ("Title 2", 2))),),
    total: 4,
  ),
)
```

```
// Two groups of titles
#headerline(
  group("Q1", "Q2", "Q3", "Q4"),
  group("Q1", "Q2", "Q3", "Q4"),
)
```

```
(
  (
    type: "header",
    headers: (
      (
        titles: (("Q1", 1), ("Q2", 1), ("Q3", 1), ("Q4", 1)),
      ),
      (
        titles: (("Q1", 1), ("Q2", 1), ("Q3", 1), ("Q4", 1)),
      ),
    ),
    total: 8,
  ),
)
```

```
// Two lines of headers
#headerline(
  group(("2023", 4), ("2024", 4))
)
#headerline(
  group("Q1", "Q2", "Q3", "Q4"),
  group("Q1", "Q2", "Q3", "Q4"),
)
```

```
(
  (
    type: "header",
    headers: ((titles: (("2023", 4), ("2024", 4))),),
    total: 8,
  ),
) (
  (
    type: "header",
    headers: (
      (
        titles: (("Q1", 1), ("Q2", 1), ("Q3", 1), ("Q4", 1)),
      ),
      (
        titles: (("Q1", 1), ("Q2", 1), ("Q3", 1), ("Q4", 1)),
      ),
    ),
    total: 8,
  ),
)
```

`#group(..titles)`

Defines a group of titles in a header line.

Takes the same options as `#headerline`.

`#task(name, style, ..lines)`

Defines a task

Argument

`name` `content`

The name of the task

Argument

`style: none` `dictionary`

The style to use for the task line. If not specified, the default style will be used.

Argument

`..lines` `array`

The lines to display in the task. Can be specified in several different formats:

```
// Spans 1 month, starting at the first month of the timeline
#task("Task", (0, 1))
```

```
(
  (
    type: "task",
    name: "Task",
    lines: ((from: 0, to: 1),),
  ),
)
```

```
// One red line at month 1, and a line spanning 2 months starting at month 4
#task("Task", (from: 0, to: 1, style: (stroke: red)), (3, 5))
```

```
(
  (
    type: "task",
    name: "Task",
    lines: (
      (from: 0, to: 1, style: (stroke: rgb("#ff4136"))),
      (from: 3, to: 5),
    ),
  ),
)
```

`#taskgroup(title, tasks)`

Groups tasks together in a box. If `title` is specified, a title will be displayed, with a line spanning all the inner tasks.

Argument

`title: none` `content`

The title of the task group

Argument

`tasks` `content`

The tasks to display in the group

```
#taskgroup(title: "Research", {
  task("Task 1", (0, 1))
  task("Task 2", (3, 5))
})
```

```
(
  (
    type: "taskgroup",
    tasks: (
      (
        type: "task",
        name: "Research",
        lines: ((from: 0, to: 5),),
      ),
      (
        type: "task",
        name: "Task 1",
        lines: ((from: 0, to: 1),),
      ),
      (
        type: "task",
        name: "Task 2",
        lines: ((from: 3, to: 5),),
      ),
    ),
  ),
)
```

```
#milestone(body,
  at,
  style,
  overhang,
  spacing,
  anchor: "top")
```

Defines a milestone. The way it's displayed depends on the `milestone-layout` option of the `#timeline` command.

Argument

`at` — `float`

The month at which the milestone should be displayed. Can be fractional.

Argument

`style: ()` — `dictionary`

Style for the milestone line. Defaults to `milestone-line-style`.

Argument

`overhang: 5pt` — `length`

How far the milestone line should extend past the end of the timeline. Defaults to `milestone-overhang`.

Argument

`spacing: 5pt` `length`

Spacing between the milestone line and the text. Defaults to `spacing`.

Argument

`anchor: "top"` `string`

The anchor point for the milestone text. Can be `top`, `bottom`, `left`, `right`, `top-left`, `top-right`, `bottom-left`, `bottom-right`, `center`, `center-left`, `center-right`, `center-top`, `center-bottom`. Defaults to `top`.

Argument

`body` `content`

The text to display next to the milestone line

Part I.

Index