As easy as pie chart

Intro

VADYM HORBONOS

Conic gradient

— це вид градієнту при якому колор стопи розміщюються по окружності кола

Conic gradient — Draft

— не увійшов до специфікації CSS 3. Зараз знаходиться у драфті CSS image module 4

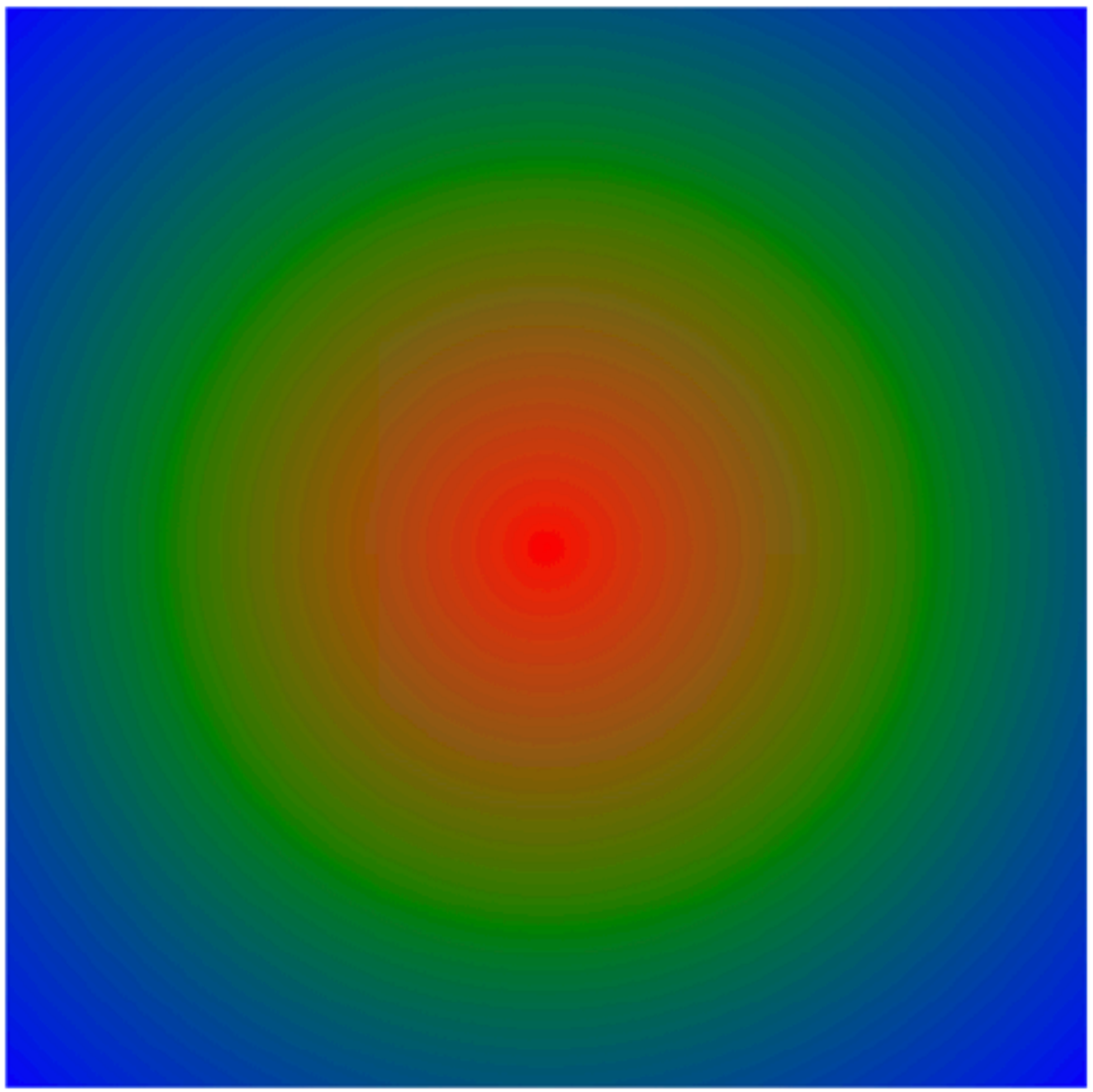
https://www.w3.org/TR/css-images-4/#conic-gradients

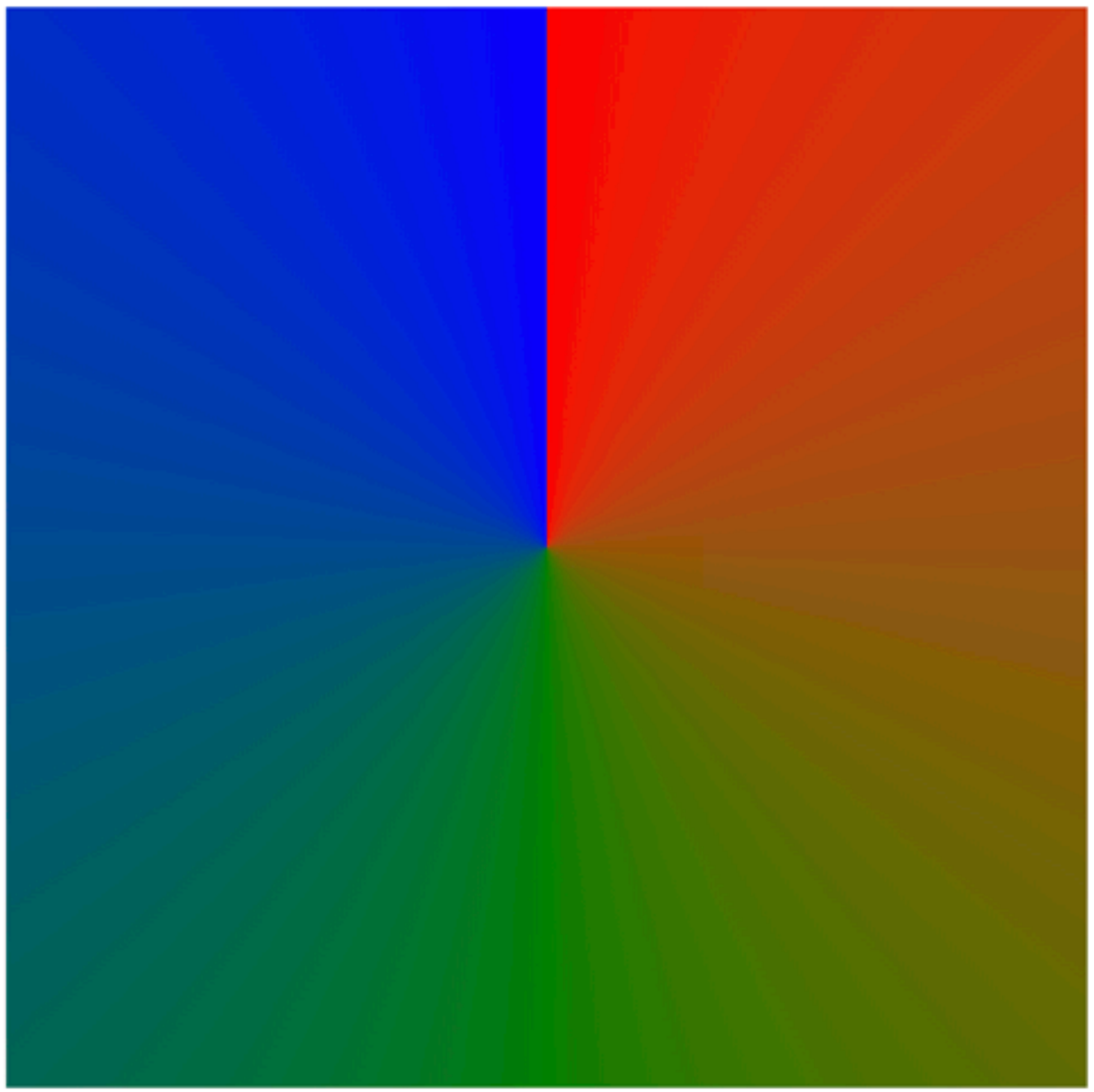
Conic gradient Evolution

Conic vs Radial

Comparison

https://codepen.io/vadym1930/pen/VwYNWKQ





Browser support

https://caniuse.com/#search=conic
https://leaverou.github.io/conic-gradient/ — polyfill

CSS Conical Gradients - WD

Usage % of all users ?

Global 79.88%

Method of defining a conical or repeating conical color gradient as a CSS image.

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baid Brow
			4-58												
			1 59-68	3.1-12	10-45	3.2-12.1									
6-10	12-18	2-71	69-78	12.1	1 46-63	12.2-13.1		2.1-4.4.4	12-12.1				4-9.2		
11	79	72	79	13	64	13.2	all	76	46	79	68	12.12	10.1	1.2	7.1
		73-74	80-82	TP		13.3									

Dynamic Pie chart

changing the CSS properties for the DOM element

Approaches

1) INLINE STYLES goes from template

2) INLINE DATA-ATTR goes from template to manipulate with JS

3) CK EDITOR PLUGIN

4) Grab data via JSON API

5) Grab data via GRAPHQL

6) DO NOT USE, go with one of the contributed modules

Approaches Why

1) INLINE STYLES goes from template

— can’t apply polyfill without preprocess, not ok

2) INLINE DATA-ATTR goes from template to manipulate with JS

— ok

3) CK EDITOR PLUGIN

— better than ok

4) Grab data via JSON API

— ok, if you do not afraid to grab paragraph inside paragraph inside paragraph …

5) Grab data via GRAPHQL

— ok

6) DO NOT USE, go with one of the contributed modules

— ok

Approaches Choice

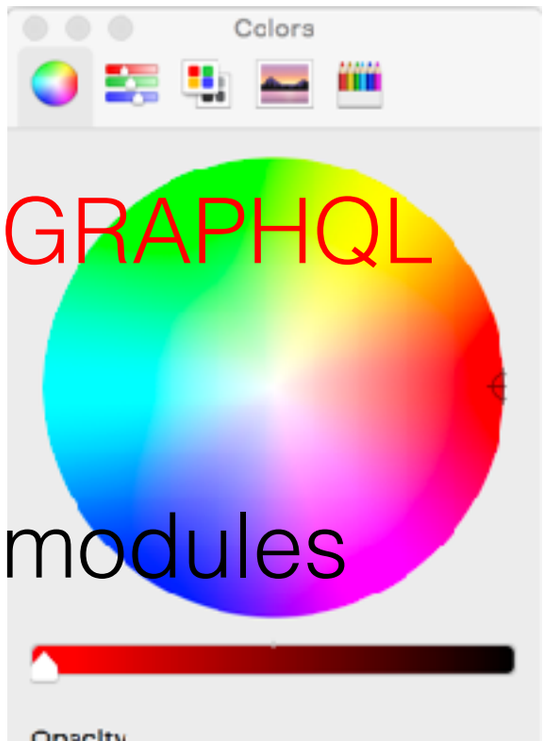
1) INLINE STYLES goes from template

2) INLINE DATA-ATTR goes from template to manipulate with JS

3) CK EDITOR PLUGIN

4) Grab data via JSON API

5) Grab data via GRAPHQL



6) DO NOT USE, go with one of the contributed modules

Basis

```
pieStyles() {
      let acum = 0;
      let styles = [
         { color: "#00A37A", value: 40 },
         { color: "#365164", value: 30 },
         { color: "#a54f93", value: 30 }
      ].map(
        segment => `${segment.color} 0 ${(acum += segment.value)}%`
      );

      return {
        background: `conic-gradient( ${styles.join(",")} )`
      };
  }
```

https://vuedose.tips/tips/the-most-modern-pie-chart-component-using-css-conic-gradient-and-vue-js/

Helper

```
/**
 * Generates comma separated string with color stops values
 * Accept an array of objects { color: 'valid color', value: number }
 *
 * @param {array} arr
 *
 * @returns string
 */
generateStyle(arr = []) {
  if (!arr[Symbol.iterator]) {
    arr = [];
  }

  let acum = 0;
  let styles = arr.map(
    segment => `${segment.color} 0 ${(acum += Number(segment.value))}%`
  );

  return styles.join(",");
}
```

План

1. Отримати дані через Graphql
2. Адаптувати дані
3. Написати юніт тести
4. Переконатися, що це працює для всіх параграфів на сторінці
5. Додати поліфіл

План підготовчих дій

1. Продумати заповнення контенту та сеттінгів для пай чартів
2. Підключити необхідні бібліотеки, налаштувати зібрання бандлу

Pie Chart

Pie Chart

Remove

Description

Source

Drupal Content

Пай чарт в адмін панелі може мати вигляд параграфа

body p

About text formats

Show row weights

Labels

Label one

Label two

Add another item

Show row weights

Values

55

45

Add another item

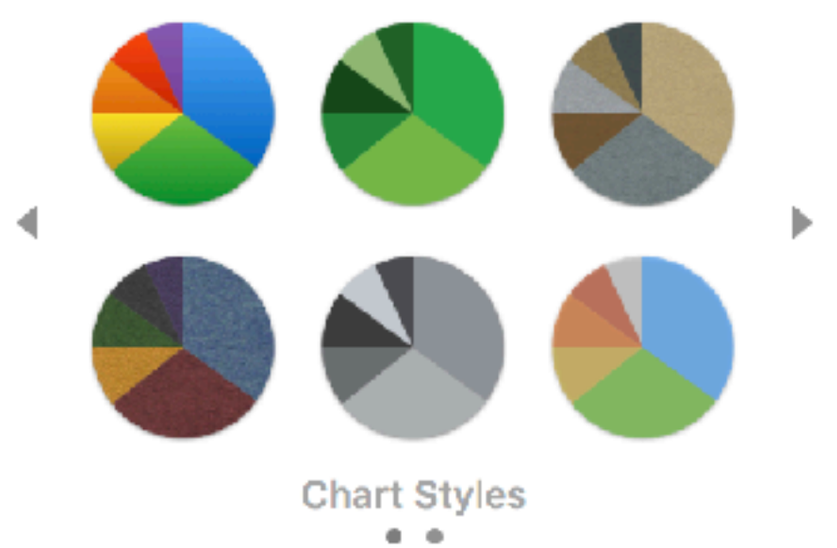
Show row weights

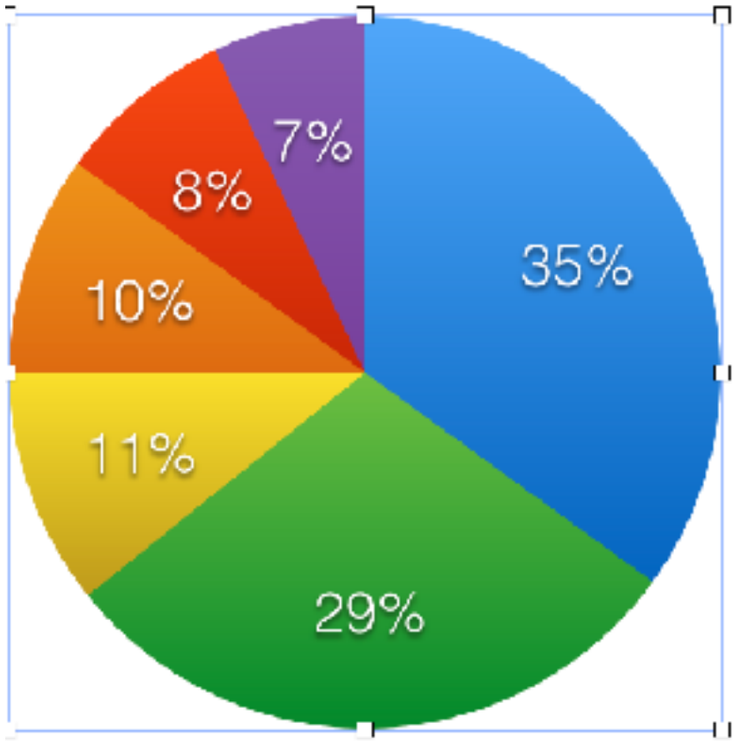
Colors

crimson

lime

Ideal Settings look something like this





CK Editor plugin could look like this

Edit Chart Data

Chart Data

	April	May	June	July	August	September
Region 1	91	76	28	26	21	18

Graphql query

GraphiQL ▶ Prettify History

```
query ($nid: String!) {
  nodeQuery (filter: {conditions: [
  {operator: EQUAL, field: "type", value: ["page"]}
  {operator: EQUAL, field: "nid", value: [$nid]}
]}) {
    count
    entities {
      ...on NodePage{
        nid
        title
        fieldPieChart {
          entity{
            ... on ParagraphPieChart {
              fieldColors
              fieldLabels
              fieldDescription {
                processed
              }
              fieldValues
            }
          }
        }
      }
    }
  }
}
```

QUERY VARIABLES

```
{
  "nid": "1"
}
```

Graphql response


```
{
  "data": {
    "nodeQuery": {
      "count": 1,
      "entities": [
        {
          "nid": 1,
          "title": "Title",
          "fieldPieChart": [
            {
              "entity": {
                "fieldColors": [
                  "crimson",
                  "lime"
                ],
                "fieldLabels": [
                  "Label one",
                  "Label two"
                ],
                "fieldDescription": {
                  "processed": "<p>Description</p>\n"
                },
                "fieldValues": [
                  "55",
                  "45"
                ]
              }
            },
            {
              "entity": {
                "fieldColors": [
                  "blue",
                  "pink"
                ],
                "fieldLabels": [
                  "Label one",
                  "Label two"
                ],
                "fieldDescription": {
                  "processed": "<p><strong>One more chart</strong></p>\n"
                },
                "fieldValues": [
                  "30",
                  "70"
                ]
              }
            }
          ]
        }
      ]
    }
  }
}
```


1

2

Two entities available

Drupal integration

Варіант гібриду

— Рутінг, безпека, контент регулюються Drupal.

— Вигляд сторінок повністю керовані JavaScript бібліотекою.

— Для клієнту гібриду підійжуть легкі, гнучкі бібліотеки. Не має особливої різниці, якщо ви виберете React або Vue.js або Svelte.

У якості бонусу ви можете отримати досить швидкий перший рендер, якщо передасте з бекенду всі дані для дефольтного рендеру сторінки

node--page.html.twig

docroot > themes > [redacted] > templates > node--page.html.twig

```
20
21  {{ attach_library('draft_theme/js-app') }} <!— Підключаємо бібліотеку —>
22
23  <script>
24    window.nodeIdForPieChart = "{{ node.id }}";
25  </script>   <!— Передаємо у глобальний скоуп id ноди (там міг бути весь контент
                сторінки в json) —>
26
27    <div{{ content_attributes }}>
28      <div id="app"></div>
29    </div>
30
31  </article>
32
```

Hybrid

Libraries organisation

```
18
19  js-vendors:
20    js:
21      pie-chart-app/dist/chunk-vendors.js: {}
22
23  js-app:
24    css:
25      theme:
26        pie-chart-app/dist/app.css:
27    js:
28      pie-chart-app/dist/app.js: {}
29    dependencies:
30      - adminimal_theme/js-vendors
31      - adminimal_theme/conic-polyfill-prefix
32      - adminimal_theme/conic-polyfill-gradient
33
34  conic-polyfill-prefix:
35    js:
36      pie-chart-app/shared/prefix-free.js: {}
37
38  conic-polyfill-gradient:
39    js:
40      pie-chart-app/shared/conic-gradient.js: {}
41
```

main JS vendors

main CSS

main JS

polyfills

View Edit Delete Revisions

Home

Member for 2 weeks

Submitted by admin on Sun, 01/26/2020 - 14:48

Result

Review Response on Page 18

Resume

— На основі простої CSS властивості можна побудувати один із найпопулярніших видів чартів

— При прийнятті рішення, яким чином реалізувати пай чарти варто враховувати

1. На скільки кастомізованим необхідний кінцевий результат
2. Чи необхідні інші види чартів у майбутьому
3. Які ресурси ви маєте (бекенд, фронтенд)
4. Чи готові написати тести для кастомного рішення
5. Інші

— Загальне правило не використовувати кастомних рішень, крім випадків коли не має можливості застосувати готове рішення

— Не виключено, що бібліотека для пай чартів під капотом використовує подібну технологію

— Експериментуючи ми більше розуміємо як працюють речі, що дозволяє свідоміше використовувати готові рішення

Resources

1. https://www.w3.org/TR/css-images-4/#conic-gradients
2. https://css-tricks.com/snippets/css/css-conic-gradient/
3. https://developer.mozilla.org/en-US/docs/Web/CSS/conic-gradient
4. https://leaverou.github.io/conic-gradient/
5. Check out client app source code https://github.com/vadym1930/pie-chart-vue