

## # 高品質インフォグラフィック HTML スライド作成プロンプト

以下の指示に従い、与えられたテキストを\*\*高品質なインフォグラフィック HTML\*\*に変換すること。

**目的：**高い情報密度と優れた視認性を両立し、エグゼクティブプレゼンにも耐えうるプロフェッショナルなビジネススタイルを実現すること。

### ## 1. 全体設計

#### ### 1.1 基本仕様

- \* **スライド形式：**1 枚スライド (16:9) であること。
- \* **レスポンス対応：**横幅 100%、最大幅 1600px であること。
- \* **レイアウト：**中央寄せ、情報量に応じて 1, 2, 3 カラムを自動切り替えとすること。
  - \* 1 カラム：幅 100% とすること。
  - \* 2 カラム：各カラム 50% とすること。
  - \* 3 カラム：各カラム 33% とすること。
- \* **カラム間隔：**24px とすること。
- \* **情報フロー：**上から下への自然な流れとすること。
- \* **フッター：**ページ下部に出典情報を明記すること。
- \* **グラフ挿入：**適切な箇所にグラフを挿入できる設計とすること。
- \* **スクロールバー：**可能な限りスクロールバーを表示させないこと。やむを得ず表示する場合は、最小限にとどめ、デザインを損なわないように配慮すること。

#### ### 1.2 デザイン指針

- \* **視覚的階層：**情報の重要度に応じた明確な階層構造であること。
- \* **統一感：**全体を通して一貫性のあるスタイルであること。
- \* **簡潔性：**過剰な装飾を避け、情報を効果的に伝えること。
- \* **訴求力：**記憶に残りやすく、説得力のあるビジュアルであること。
- \* **余白：**適切に余白を設け、圧迫感を軽減すること。
- \* **情報構造：**論理的で一貫性のある垂直方向の流れであること。
- \* **親しみやすさとプロフェッショナル感：**手書き風要素(罫線、矢印)で親しみやすさを演出しつつ、洗練されたデザインでプロフェッショナルな印象を与えること。

### ## 2. デザイン仕様

#### ### 2.1 カラースキーム

- \* **メインテキスト：**`#334155` とすること。
- \* **見出し：**`#EE6C8A` (Flory Pink) とすること。
- \* **サブタイトル：**`#475569` とすること。
- \* **背景：**白ベース (`#ffffff`) またはグラデーション (`linear-gradient(135deg, #f8fafc 0%, #f1f5f9 100%)`) とすること。
- \* **カード背景：**`#ffffff` (白) とすること。
- \* **追加カラーパレット：**

<palette>

<color name=' ビジネス-1' rgb=' 0A2463' r=' 10' g=' 36' b=' 99' />

<color name=' ビジネス-2' rgb=' 1E56A0' r=' 30' g=' 86' b=' 160' />

<color name=' ビジネス-3' rgb=' 3D84B8' r=' 61' g=' 132' b=' 184' />

<color name=' ビジネス-4' rgb=' 78A6C8' r=' 120' g=' 166' b=' 200' />

<color name=' ビジネス-5' rgb=' E6F2F5' r=' 230' g=' 242' b=' 245' />

</palette>

...

とすること。

### ### 2.2 タイポグラフィ

- \* \*\*フォント\*\* : \*\* Noto Sans JP, M PLUS 1p, Shippori Mincho を使用すること。

...

<style>

@import

url('https://fonts.googleapis.com/css2?family=Noto+Sans+JP:wght@400;500;700&family=M+PLUS+1p:wght@400;500;700&family=Shippori+Mincho:wght@400;500;700&display=swap');

</style>

...

- \* \*\*要素別スタイル\*\* :

- \* \*\*タイトル\*\* : \*\* 36px, `#0A2463`, 太字, シャープな印象とすること。
- \* \*\*サブタイトル\*\* : \*\* 22px, `#1E56A0`, セミボールドとすること。
- \* \*\*セクション見出し\*\* : \*\* 24px, `#3D84B8`, アイコン付きとすること。
- \* \*\*本文\*\* : \*\* 16px, `#333333`, 行間 1.6 とすること。

### ### 2.3 レイアウト要素

- \* \*\*ヘッダー\*\* : \*\* 中央揃えタイトル + 右揃え日付/出典とすること。
- \* \*\*カラム構成\*\* : \*\* 情報量に応じて 1, 2, 3 カラムを自動で切り替えること。
- \* \*\*カード\*\* : \*\* 白背景 (`#ffffff`), 角丸 10px, シャドウとすること。
- \* \*\*セクション区切り\*\* : \*\* 区切り線または背景色 (`#E6F2F5`) とすること。
- \* \*\*その他\*\* : \*\* 情報密度に応じた適切な余白と階層構造とすること。
- \* \*\*レスポンシブ対応\*\* : \*\* 横幅 100%、最大幅 1600px を維持すること。

## ## 3. グラフィックレコーディング表現

- \* \*\*情報構造\*\* : \*\* 重要度に基づいた垂直方向の情報構造とすること。
- \* \*\*セクション\*\* : \*\* 各セクションは独立したまとまりとして表現すること。
- \* \*\*強調\*\* : \*\*
  - \* キーワードを青系の色や太字で強調すること。
  - \* `highlight` クラスを適用 (`#fdf2f4` の背景色) すること。
  - \* ポイントや見出しに絵文字 (👉, ☆ など) を使用すること。
- \* \*\*視覚化\*\* : \*\*
  - \* プロフェッショナルなアイコン (📊, 📅 など) を使用すること。
  - \* 幾何学的な図形、グラフ、図表でデータを表現すること。
  - \* 手書き風の要素 (罫線、矢印) を使用すること。

## ## 4. コンポーネント設計と CSS

### ### 4.1 基本コンポーネント

- \* \*\*カード\*\* : \*\* タイトル、コンテンツ (テキスト、リスト、テーブル、グラフなど) とすること。

...

.card {

background-color: white;

border-radius: 10px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);

padding: 12px;

height: auto;

```
    overflow: hidden;
    border: 1px solid #e2e8f0;
    display: flex;
    flex-direction: column;
  }
  ...
```

### ### 4.2 レイアウト設計

\* \*\*ベースコンテナ:\*\*

```
...
.slide-container {
  width: 1600px;
  height: 900px;
  background-color: white;
  border-radius: 12px;
  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.08);
  padding: 32px;
  position: relative;
  overflow: hidden;
}
...
```

\* \*\*ヘッダー:\*\*

```
...
.header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 20px;
  border-bottom: 1px solid #e2e8f0;
  padding-bottom: 16px;
}
...
```

\* \*\*可変カラム構成 (1, 2, 3 カラム対応):\*\*

```
...
.content {
  display: flex;
  height: calc(100% - 80px);
  gap: 24px;
  /* カラム数を動的に変更するための設定 */
  flex-wrap: wrap;
  justify-content: flex-start; /* 左寄せ */
}

.column {
  height: 100%;
  display: flex;
  flex-direction: column;
  gap: 16px;
  overflow: hidden;
}
```

```

/* カラム数を動的に変更 */
}

/* 1 カラムの場合 */
.one-column {
    width: 100%;
}
.one-column .column {
    width: 100%;
}

/* 2 カラムの場合 */
.two-column {
    width: calc(100% + 24px); /* 相殺 */
    display: flex;
    flex-wrap: wrap;
}
.two-column .column {
    width: calc(50% - 12px); /* ギャップを考慮 */
}

/* 3 カラムの場合 */
.three-column {
    width: calc(100% + 24px); /* 相殺 */
    display: flex;
    flex-wrap: wrap;
}
.three-column .column {
    width: calc(33.333% - 16px); /* ギャップを考慮 */
    margin-right: 8px;
}
...

```

\* \*\*上記 CSS の使用方法:\*\*

\* コンテンツの量に応じて、`content` クラスを持つ要素に `one-column`、`two-column`、`three-column` のいずれかのクラスを追加することで、カラム数を制御します。HTML 側で、例えば`<div class="content two-column">`のように指定します。

### ### 4.3 スタイル調整

\* \*\*オーバーフロー対策:\*\*

\* スクロールバーを非表示にするために、`overflow: hidden;` を基本とし、必要に応じてコンテンツを調整すること。  
 \* どうしてもスクロールが必要な場合は、`overflow-x: auto;` または `overflow-y: auto;` を使用し、最小限にとどめること。

```

...
.card-content {
    font-size: 13px;
    overflow-wrap: break-word;
    overflow: hidden; /* または auto, 必要に応じて調整 */
    max-height: 100%;
}
...

```

\* \*\*リスト要素最適化:\*\*

```
...  
ul {  
  padding-left: 16px;  
  margin-bottom: 6px;  
}  
  
li {  
  margin-bottom: 4px;  
  font-size: 12px;  
  line-height: 1.3;  
}  
...
```

\* \*\*ハイライト要素:\*\*

```
...  
.highlight {  
  background-color: #fdf2f4;  
  padding: 2px 4px;  
  border-radius: 4px;  
  font-weight: 500;  
}  
...
```

#### ### 4.4 拡張コンポーネント

\* \*\*プロセスフロー:\*\*

```
...  
.process-flow {  
  display: flex;  
  justify-content: space-between;  
  margin: 16px 0;  
  position: relative;  
}  
  
.process-step {  
  text-align: center;  
  width: 22%;  
  z-index: 2;  
}  
  
.process-icon {  
  background-color: #EE6C8A;  
  color: white;  
  width: 36px;  
  height: 36px;  
  border-radius: 50%;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  margin: 0 auto 8px;  
  font-size: 16px;
```

```
    font-weight: bold;
}
```

```
.process-title {
    font-size: 13px;
    font-weight: 600;
    margin-bottom: 4px;
}
```

```
.process-desc {
    font-size: 11px;
}
...
```

\* \*\*グリッドレイアウト:\*\*  
...

```
.capability-grid {
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    gap: 12px;
    margin-top: 12px;
    border: 2px solid #e2e8f0; /* グリッド全体の枠線を太くする */
    padding: 10px; /* 必要に応じて内側の余白を追加 */
}
```

```
.capability-item {
    display: flex;
    align-items: center;
    gap: 8px;
    border: 1px solid #e2e8f0; /* 各アイテムの枠線 */
    padding: 5px;
}
```

```
.capability-icon {
    font-size: 14px;
    color: #EE6C8A;
    min-width: 20px;
}
...
```

\* \*\*アイコンリスト:\*\*

...

```
.integration-list {
    display: grid;
    grid-template-columns: repeat(5, 1fr);
    gap: 6px;
    margin-top: 8px;
}
```

```
.integration-item {
    background-color: #fdf2f4;
    border-radius: 12px;
    padding: 3px 6px;
    font-size: 11px;
}
```

```

display: flex;
align-items: center;
gap: 2px;
justify-content: center;
white-space: nowrap;
overflow: hidden;
text-overflow: ellipsis;
}
...

```

\* \*\*テーブル:\*\*

```

...

.comparison-table {
width: 100%;
border-collapse: collapse;
margin-top: 10px;
font-size: 12px;
border: 2px solid #e2e8f0; /* テーブル全体の枠線を太くする */
}

.comparison-table th,
.comparison-table td {
padding: 8px;
border: 1px solid #e2e8f0; /* セルの枠線 */
}

.comparison-table th {
background-color: #fdf2f4;
text-align: left;
}
...

```

## ## 5. 実装のベストプラクティス

### ### 5.1 情報最適化

- \* 長文は要点を絞って圧縮（1文20-30字程度）すること。
- \* 箇条書きは簡潔に（1-2行）すること。
- \* 重要概念を選別し、不要な詳細は削除すること。

### ### 5.2 テキスト調整

- \* タイトルは簡潔かつ的確に内容を表すこと。
- \* 複合語は短縮すること。
- \* 体言止めを積極的に活用すること。
- \* 接続詞・副詞は削減すること。

### ### 5.3 視覚的階層

- \* 情報の重要度に応じた視覚的なヒエラルキーを確立すること。
- \* 強調したいポイントには `highlight` クラスを適用すること。
- \* 補足情報はより小さなフォントサイズで表示すること。
- \* 視認性の高いアイコンを選定すること。

## ## 6. HTML 構造（ひな形）

```
```html
<!DOCTYPE html>
<html lang=""ja"">
<head>
  <meta charset=""UTF-8"">
  <meta name=""viewport"" content=""width=device-width, initial-scale=1.0"">
  <title>インフォグラフィックスライド</title>
  <style>
    /* ここにスタイルを定義 */
  </style>
</head>
<body>
  <div class=""slide-container"">
    <div class=""header"">
      <h1 class=""header-title"">メインタイトル</h1>
      <div class=""header-date"">YYYY 年 MM 月</div>
    </div>

    <div class=""content"">
      <!-- 左カラム -->
      <div class=""column"">
        <div class=""card"">
          <div class=""card-title""> セクション 1</div>
          <div class=""card-content"">
            コンテンツ
          </div>
        </div>

        <!-- 他のカード -->
      </div>

      <!-- 右カラム -->
      <div class=""column"">
        <!-- カード -->
      </div>
    </div>

    <div class=""footer"">フッター情報 | © YYYY</div>
  </div>
</body>
</html>
```
```

## ## 7. 出力時の注意点

- 文字のはみ出しが最も重要な問題。絶対に発生させないこと
- 特にリスト、テーブル、長文パラグラフには注意
- コンテンツ量が多い場合は、フォントサイズを 12px まで小さくする
- グリッド/テーブルのセル幅を調整し、均等配分より内容に合わせた配分を優先
- スクロールバーが必要な場合は、美観を損なわないよう最小限に
- カード間のバランスを考慮し、高さの極端な偏りを避ける