

وراثت (Inheritance) در CSS

وراثت در CSS یک از مفاهیم بنیادی است که تعیین می‌کند چگونه برخی ویژگی‌های CSS را از عناصر والد به فرزند منتقل می‌شود. این مکانیزم به توسعه‌دهندگان کمک می‌کند تا برای تغییرات تکراری، آسان‌تر و سریع‌تر بنویسند.

نحوه کار وراثت در CSS

وقتی یک ویژگی CSS برای یک عنصر والد تعریف می‌شود، فرزند آن می‌تواند آن ویژگی را به ارث ببرد، تغییر دهد یا حتی آن را حذف کند. هر یک از این فرزند مقدار متفاوتی تعریف شده، به این شکل:

```
body {
```

```
font-family: Arial,
```

```
font-size: 2rem
```

```
color: #aaa
```

در این مثال، ما داخل `<body>` (به استثنای مواردی که به طور خاص استایل متفاوتی دارند) نوشتیم `font-size`

ویئرل کی فائل آرٹس

بہار، چھترن، ویئرل کی فائل آرٹس پر مشتمل

font-family

color

line-height

text-align

visibility

cursor

ویئرل کی فائل آرٹس

margin

padding

border

width/height

background

position

Controlling

Inheritance

میراثہ

inherit

استفادہ از میراث

کونٹرولنگ، ویئرل کی فائل آرٹس پر (بکریٹر کے حوالہ)
والد بکریٹر

child

padding: inherit;

۲- استفاده از initial

مقدار پیش فرض رو رنگ را اعمال می کند.

child {

color: initial;

}

۳- استفاده از unset

۱- ابتدا به افعی ویرن قابل از رنگی برل اندک، آخر، رفتار

برل ویرن کل قابل

child {

color: unset;

border: unset;

}

احمد وراثت در طراح

۱- داشتن تکرار در نیاز به توقف مجدد ویرن کل متنا

برل نام ویرن مندرت

۲- متنا طراح در تغییرات در نحوه والدین صورت خود را

فرزندان منتقل می شود

۳- سازماندهی بجز در رنگ و در یک بین می را آسان می کند

تفاوت در زبان های برنامه نویسی

میراثی در C++ : مفهومی که در زبان C++ نیز وجود دارد

تفاوت در

ارتباط وراثتی از والد به فرزند (در هر دو)

تفاوت بازنویسی (override) در هر دو موجود است

تفاوت در نحوه عمل

تفاوت در کلیت

OOP	CSS	ویرس
باز منطبق و ساختار	باز منطبق (Presentation)	نوعی باز
کاملاً قابل برنامه ریزی	حدود و از پیش تعریف شده	تغییرپذیر و دینامیک
بر زبان های خود را می سازد	تک و والد	خودمانند
توانع و محدودیت وجود دارد	وجود ندارد	محدود
در برخی زبان ها وجود دارد	موجود	ایستادن
تکامل از OOP	موجود	تکامل

حداقل تفاوت از CSS مثل OOP

معمولاً در اینجا

در CSS می‌توانیم تگ‌ها را به صورت والد، پسر

در OOP یعنی در کلاس‌ها، را به عنوان آنتی بک

۳- عموماً وجود مقدار وراثت را

۴- عموماً امکان اُورراید (override)

در CSS عموماً ما فرزند، اجبار و میراث را به ارث

مُراس (override) می‌دهیم

در OOP می‌توانیم آنتی بک تگ‌ها را به ارث

۵- عموماً وجود کپی‌سازی

در CSS عموماً آنتی بک تگ‌ها را به ارث

در OOP می‌توانیم آنتی بک تگ‌ها را به ارث Private/protected

معمولاً در OOP

در CSS