

# SELECTROMANCY

A Grimoire of CSS Selector Spells

# A spell only works when it reaches its true target.

Welcome, apprentice. Before gradients shimmer or animations roar to life, we must master **Selectromancy**. **Selectromancy** — the subtle magic of finding exactly the right element. It is the foundation of purposeful, readable, and resilient CSS.

**WHERE?:**  
Where in the DOM?  
(Descendants,  
children, siblings)



**WHO?:**  
Which elements?  
(Tags, classes, IDs)

**WHEN?:**  
In what state?  
(Hover, focus, valid)

**HOW?:**  
How are they grouped?  
(First, last, nth-of-type)

```
/* Aim at every story card */  
.story { ... }
```

```
/* Aim at the featured one only */  
.story.is-featured { ... }
```

# The Five Schools of Selector Magic

Our journey will lead us through the great schools of the Selectromancy Academy. Each holds a different key to mastering the DOM.



**🏛️ Familial Sorcery:**  
Reading the branches of  
the DOM — parents,  
children, and siblings.



**🔮 Relational Runes:**  
The order and lineage of  
elements — first, last,  
and nth.



**⚡ States & Spirits:**  
How elements respond to  
interaction and reveal  
their state.



**➊ Arcane Attributes:**  
Reading the runes etched  
directly into HTML  
metadata.

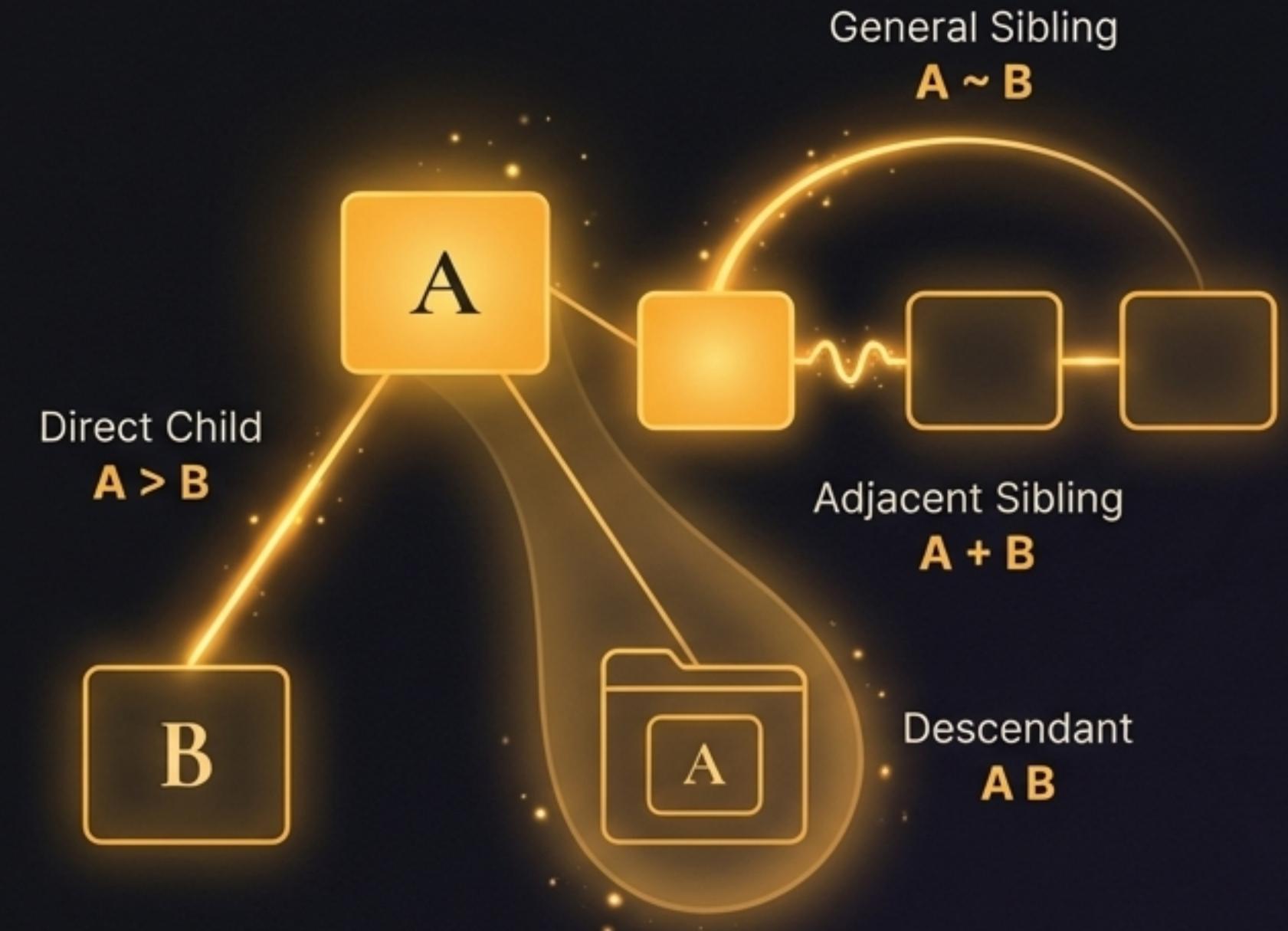


**🌌 The Unconditional Arts:**  
The high runes of logic —  
:is(), :not(), :has(),  
and :where().

# SCHOOL I: FAMILIAL SORCERY

## Reading the DOM Family Tree

Every spell begins with a question: “Where does this element live in the family tree?” This school teaches us to read that tree using **combinators** — the runes that express relationships.



# Spellwork: The Hall of Stories

THE TARGET



THE INCANTATIONS

On the Nature of Siblings  
Arcane CSS · 3 min read

Children of the Root  
DOM Lore · 2 min read

On the Nature of Siblings  
Arcane CSS · 3 min read

Children of the Root  
DOM Lore · 2 min read

Arcane CSS · 3 min read

Chiidren of the Root  
DOM Lore · 2 min read

Further tales are still being inscribed...

## Incantation 1: Descendant

Goal: Style *every* story card inside the hall.

```
.hall-of-stories .story { /* styles */ }
```

## Incantation 2: Direct Child

Goal: Style *only* the title that is a direct child.

```
.story > .story-title { /* green glow */ }
```

## Incantation 3: Adjacent Sibling

Goal: Style the meta line *immediately after* a title.

```
.story-title + .story-meta { /* amber styles */ }
```

## Incantation 4: General Sibling

Goal: Style *everything after* the main hall title.

```
.hall-title ~ * { /* purple divider */ }
```

## SCHOOL II: RELATIONAL RUNES

# The Order and Lineage of Elements

Familial Sorcery told us *who* is related. Relational Runes ask, “**Where does this element sit among its siblings?**” Here we learn to count.

`:first-child`

`:last-child`

`:nth-child(n)` (1) (odd) (2n+1)

`:first-of-type`

`:last-of-type`

`:nth-of-type(n)`

**...child:** Counts *all* sibling elements, regardless of type.

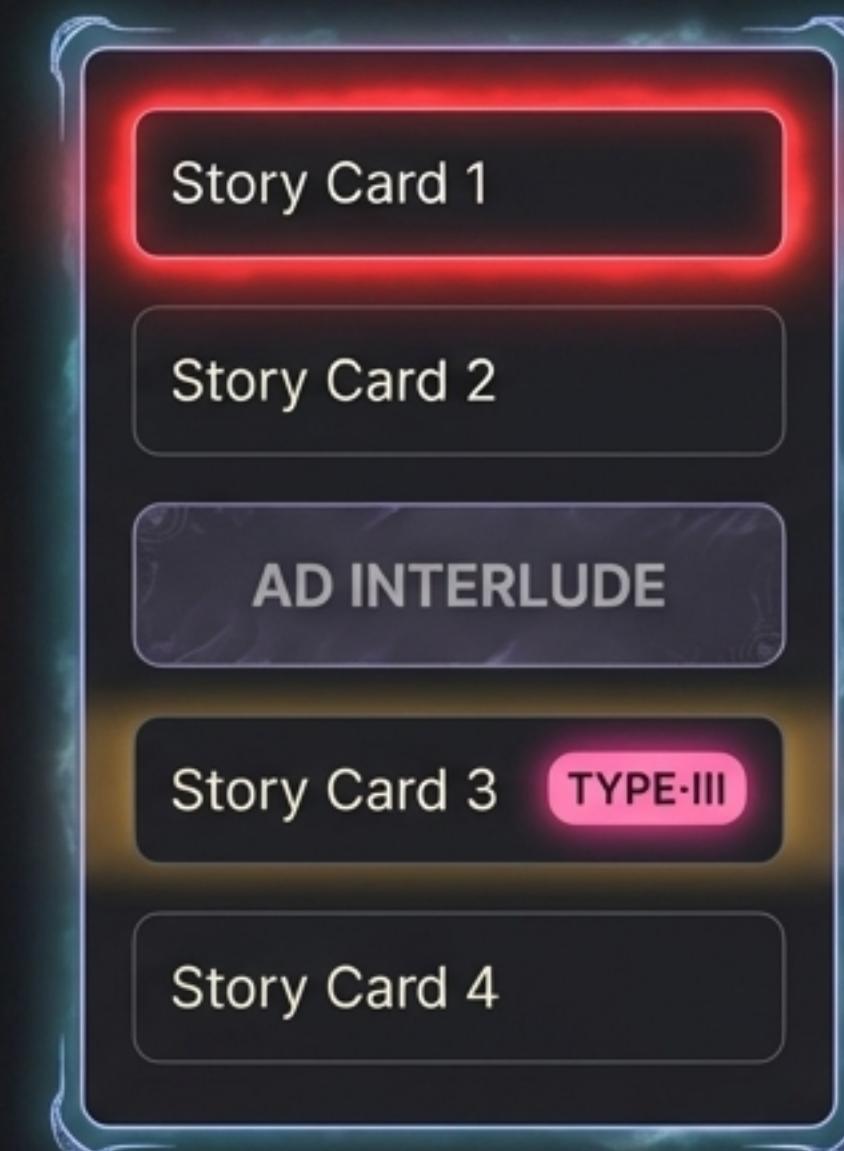
**...of-type:** Counts only siblings of the *same element type*.

# Spellwork: The Circle of Stories

## Incantation 1: The First-Born

```
.story-row:first-child {  
    /* red halo */  
}
```

Crowns the very first element in a group.



## Incantation 2: The Rhythm

```
.story-row:nth-of-type(odd) {  
    /* orange band */  
}
```

Using :nth-of-type ensures the ad doesn't break the visual rhythm of the stories.

## Incantation 3: Of Type, Not Birth

```
.story-row:nth-of-type(3) {  
    /* pink badge */  
}
```

Finds the third story article, even though it is not the third child of the parent.

## SCHOOL III: STATES & SPIRITS

# How Elements Respond, React, and Reveal State

Our spells have been about structure and position. Now, we listen for the UI's heartbeat:  
**'What happens when someone hovers, clicks, tabs, or types?'**



### INTERACTION

:hover  
:active



### FOCUS

:focus  
:focus-visible

(Note: Use `:focus-visible` for keyboard users!)



### FORM INPUT

:checked  
:disabled  
:required



### VALIDATION

:valid  
:invalid  
:placeholder-shown

# Spellwork: The Reader Preferences Panel

Spell 1: Keyboard-Aware Focus

```
.mode-btn:focus-visible {  
    /* violet outline */  
}
```

## Reader Preferences

Layout mode

Compact

Cozy

Notifications

Breaking news only

Spell 3: Intelligent Validation

```
.alerts-input:invalid:not(:placeholder-shown) {  
    /* rose border */  
}
```

Alerts Email

user@domain|

STATE

Spell 2: Parent Awakens to Child State

```
.breaking-row:has(input:checked) {  
    /* orange aura */  
}
```

Spell 4: Sibling Reaction

```
.alerts-input:valid ~ .status-pill {  
    /* lime styles */  
}
```

## SCHOOL IV: ARCANAE ATTRIBUTES

# Reading the Runes Etched into HTML

We now ask, "What secrets are already carved into the attributes?" These selectors let our CSS respond to metadata, turning HTML into a spellbook our styles can read.

### The Attribute Rune Library

Presence	<code>[attr]</code>	Has the attribute.
Exact Match	<code>[attr="value"]</code>	Attribute is exactly 'value'.
Token	<code>[attr~="token"]</code>	Attribute's space-separated list contains 'token'.
Prefix	<code>[attr^="prefix"]</code>	Value starts with 'prefix'.
Suffix	<code>[attr\$="suffix"]</code>	Value ends with 'suffix'.
Substring	<code>[attr*= "middle"]</code>	Value contains 'middle'.

## SCHOOL V: THE UNCONDITIONAL ARTS

# Where CSS Grows a Mind of Its Own

There is a deeper discipline where spellwork bends to logic itself. These runes allow selectors that reason, filter, exclude, and respond to their own contents.

### :is() - The Rune of Many Faces

Speak one spell that fits many shapes.

Combines multiple selectors that share the same styling tail.

E.g., `.card:is(.featured, .pinned) { ... }`

### :not() - The Rune of Exclusion

Declare what must be cast aside.

Styles everything except the specified element.

E.g.,  
`button:not(.primary) { ... }`

### ::has() - The Rune of Awareness

Let parents awaken to the secrets their children hold.

Styles a parent based on its descendants.

E.g.,  
`.card:has(.badge--alert) { ... }`

### ::where() - The Soft Rune

The soft incantation.

Same as :is(), but with **zero specificity**, making its rules easy to override. Perfect for resets and baselines.

E.g.,  
`.article :where(h2, p) { margin: 0; }`

# The Laws of Magical Power: Specificity

When two spells target the same element, the one with the higher 'power level,' or specificity, wins. Understanding this hierarchy prevents chaos.



The `:where()` rune always has a power level of **zero**, making its incantations the easiest to override.

# The Selectromancer's Cheatsheet

## Familial Sorcery

`section .story`  
(Descendant)

`section > .story`  
(Child)

`h2 + p`  
(Adjacent Sibling)

`h2 ~ article`  
(General Sibling)

## Relational Runes

`:first-child /  
:last-of-type`

`:nth-child(2n)`  
(Even items)

`:nth-of-type(3)`  
(Third of its kind)

## States & Spirits

`input:focus-visible`

`input:checked +  
label`

`input:valid:not(:  
placeholder-shown)`

## Arcane Attributes

`a[href^="https://"]`  
(Starts with)

`img[src$=".svg"]`  
(Ends with)

`div[data-tags~=  
"featured"]`  
(Token)

## Unconditional Arts

`:is(.card, .panel)`

`:not(.is-disabled)`

`:has(> .icon)`

`:where(h1, h2, h3)`

# Best Practices for Potent Spellcasting



## 1 Choose the Right Incantation

Our job is to choose the incantation that best balances readability, resilience, and the level of scoping we actually need.

Prefer clear, short selectors.  
Avoid overly long chains that are brittle and hard to read.



## 2 Let the Data Drive the Spell

Use attribute selectors

Use attribute selectors when the information you need is already encoded in the markup.

Don't stack on a dozen `.*is-x*` classes if `data-state="x"` already exists.

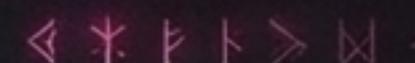


## 3 Read Your Spells Aloud

Translate selectors into plain English to test their clarity.

`.hall > .story-title` →  
“The story title that is a direct child of the hall.”

If it's hard to say, it will be hard to maintain.



p.s. > Select well. Design everything.

