Claro! Vamos analisar o seu script **linha por linha** com uma **explicação detalhada** e **exemplo passo a passo** de como ele funciona. Esse script aplica um **efeito de inclinação 3D (tilt effect)** a um elemento HTML com a classe .card conforme o movimento do mouse.

**CÓDIGO COMPLETO (para referência):**

<script>

let card = document.querySelector('.card');

card.addEventListener('mousemove', (e) => {

let rect = card.getBoundingClientRect();

let x = e.clientX - rect.left - rect.width / 2;

let y = e.clientY - rect.top - rect.height / 2;

let rotateX = (x / rect.height) \* -30;

let rotateY = (y / rect.width) \* -30;

card.style.transform = `perspective(1000px) rotate(${rotateX}deg) rotateY(${rotateY}deg)`;

let bgX = (x / rect.width) \* 50 + 50;

card.style.backgroundPositionX = `${bgX}%`;

})

card.addEventListener('mouseleave', () => {

card.style.transform = `perspective(1000px) rotateX(0deg) rotateY(0deg)`;

card.style.backgroundPositionX = ``;

})

</script>

**LINHA POR LINHA:**

**1. let card = document.querySelector('.card');**

* **O que faz:** Seleciona o elemento HTML com a classe card e o armazena na variável card.
* **Exemplo:** Se no HTML houver <div class="card">...</div>, esse elemento será manipulado pelo script.

**2. card.addEventListener('mousemove', (e) => {**

* **O que faz:** Adiciona um ouvinte de evento (mousemove) ao card.
* **Significa:** Quando o mouse se move sobre o card, a função entre { ... } será executada, recebendo o evento e.

**3. let rect = card.getBoundingClientRect();**

* **O que faz:** Obtém as dimensões e a posição atual do card na tela.
* **Exemplo:** Suponha que o card esteja a 100px da esquerda da tela e tenha 300px de largura e altura — essa informação vai estar em rect.

**4. let x = e.clientX - rect.left - rect.width / 2;**

* **O que faz:** Calcula a posição horizontal do mouse **relativa ao centro** do card.
* e.clientX → posição X do mouse na tela.
* rect.left → distância do card até a borda esquerda da tela.
* rect.width / 2 → metade da largura do card (para achar o centro).
* **Resultado:** x será um valor negativo se o mouse estiver à esquerda do centro, positivo se estiver à direita.

**5. let y = e.clientY - rect.top - rect.height / 2;**

* **Semelhante ao anterior**, mas no eixo vertical.
* **Resultado:** y será negativo se o mouse estiver acima do centro, positivo se estiver abaixo.

**6. let rotateX = (x / rect.height) \* -30;**

* **Calcula o grau de rotação em torno do eixo X** (horizontal), proporcional à posição horizontal do mouse.
* **Multiplica por -30** para dar um efeito visual mais expressivo.
* **Exemplo:** Se x = 100 e rect.height = 300, então rotateX = (100/300) \* -30 = -10.

**7. let rotateY = (y / rect.width) \* -30;**

* **Idêntico à lógica acima**, mas para o eixo Y (vertical).
* Inverte a direção também com -30.

**8. card.style.transform = \perspective(1000px) rotate(${rotateX}deg) rotateY(${rotateY}deg)`;`**

* **Aplica o efeito de rotação 3D** no card.
* perspective(1000px) cria uma sensação de profundidade (quanto menor o valor, mais dramática a perspectiva).
* rotate(...) rotateY(...) rotaciona o card nos eixos X e Y conforme o movimento do mouse.

**9. let bgX = (x / rect.width) \* 50 + 50;**

* **Calcula a posição do background no eixo X** (para criar um leve movimento no plano de fundo).
* O valor resultante varia entre 0% e 100% aproximadamente.
* Multiplica por 50 e soma 50 para centralizar o efeito em 50%.

**10. card.style.backgroundPositionX = \${bgX}%`;`**

* **Aplica o deslocamento horizontal no fundo do card.**
* Dá um efeito parallax sutil (como se o fundo também se movesse).

**11. card.addEventListener('mouseleave', () => {**

* **Quando o mouse sai do card**, a função abaixo é executada.

**12. card.style.transform = \perspective(1000px) rotateX(0deg) rotateY(0deg)`;`**

* **Restaura o card para sua posição original** (sem rotação) quando o mouse sai.

**13. card.style.backgroundPositionX = \`;`**

* **Remove a posição personalizada do background**, voltando ao padrão (centralizado ou como definido no CSS).

**EXEMPLO PASSO A PASSO (simulação):**

Suponha que:

* O card tenha 300px de largura e altura.
* O mouse está em e.clientX = 450 e e.clientY = 250.
* O card começa em rect.left = 300, rect.top = 100.

**Passo 1 – calcular x e y:**

x = 450 - 300 - 150 = 0 // mouse está no centro horizontal

y = 250 - 100 - 150 = 0 // mouse está no centro vertical

**Passo 2 – calcular rotações:**

rotateX = (0 / 300) \* -30 = 0

rotateY = (0 / 300) \* -30 = 0

O card não gira — está centralizado.

**Se o mouse vai para o canto inferior direito (e.clientX = 600, e.clientY = 400):**

x = 600 - 300 - 150 = 150

y = 400 - 100 - 150 = 150

rotateX = (150 / 300) \* -30 = -15

rotateY = (150 / 300) \* -30 = -15

Agora o card gira 15° para trás e para o lado — efeito de inclinação 3D!

Se quiser, posso criar um exemplo completo com HTML e CSS para testar isso visualmente. Deseja isso?