

# GABRIELE ROMANATO

Menu

## JavaScript: calcolare il numero di indirizzi IP disponibili in una subnet

Nell'ambito delle reti informatiche, la suddivisione di una rete IP in sotto-reti più piccole, o subnetting, è una pratica fondamentale per gestire efficacemente l'allocazione degli indirizzi IP e migliorare la sicurezza e l'efficienza della rete. Un aspetto cruciale di questa pratica è il calcolo del numero di indirizzi IP disponibili in una subnet. In questo articolo, esploreremo come implementare questa funzionalità utilizzando JavaScript, fornendo uno script che può essere facilmente integrato o modificato per vari scopi di gestione della rete.

Prima di immergerci nel codice, è importante comprendere alcune basi del subnetting. Un indirizzo IP è composto da due parti: l'identificatore della rete e quello dell'host. La maschera di sottorete, spesso espressa in notazione CIDR come /24, determina come l'indirizzo IP è diviso.

La formula per calcolare il numero di indirizzi IP in una subnet è:

Numero di indirizzi =  $2^{(32 - \text{lunghezza della maschera})}$

Questo numero include l'indirizzo di rete e l'indirizzo di broadcast, che non sono utilizzabili per i dispositivi individuali.

Ecco un esempio di funzione JavaScript che calcola il numero di indirizzi IP utilizzabili in una subnet data la notazione CIDR:

```
function calcIPAddresses(cidr) {
  // Estrai la lunghezza della maschera dalla notazione CIDR
  const maskLength = parseInt(cidr.split('/')[1], 10);

  // Calcola il numero totale di indirizzi nella subnet
  const totalAddresses = Math.pow(2, 32 - maskLength);

  // Sottrai 2 per escludere l'indirizzo di rete e di broadcast
  return totalAddresses - 2;
}
```

Per utilizzare questa funzione, si può semplicemente passare la notazione CIDR della subnet di interesse:

```
console.log(calcIPAddresses("192.168.1.0/24")); // Output: 254
console.log(calcIPAddresses("192.168.1.0/30")); // Output: 2
```

In conclusione, il calcolo del numero di indirizzi IP disponibili in una subnet è essenziale per la pianificazione e la gestione delle reti. L'implementazione di questa funzionalità in JavaScript permette agli sviluppatori e agli amministratori di rete di integrare facilmente questa capacità nei loro strumenti e dashboard personalizzati.

## Applicazioni Correlate

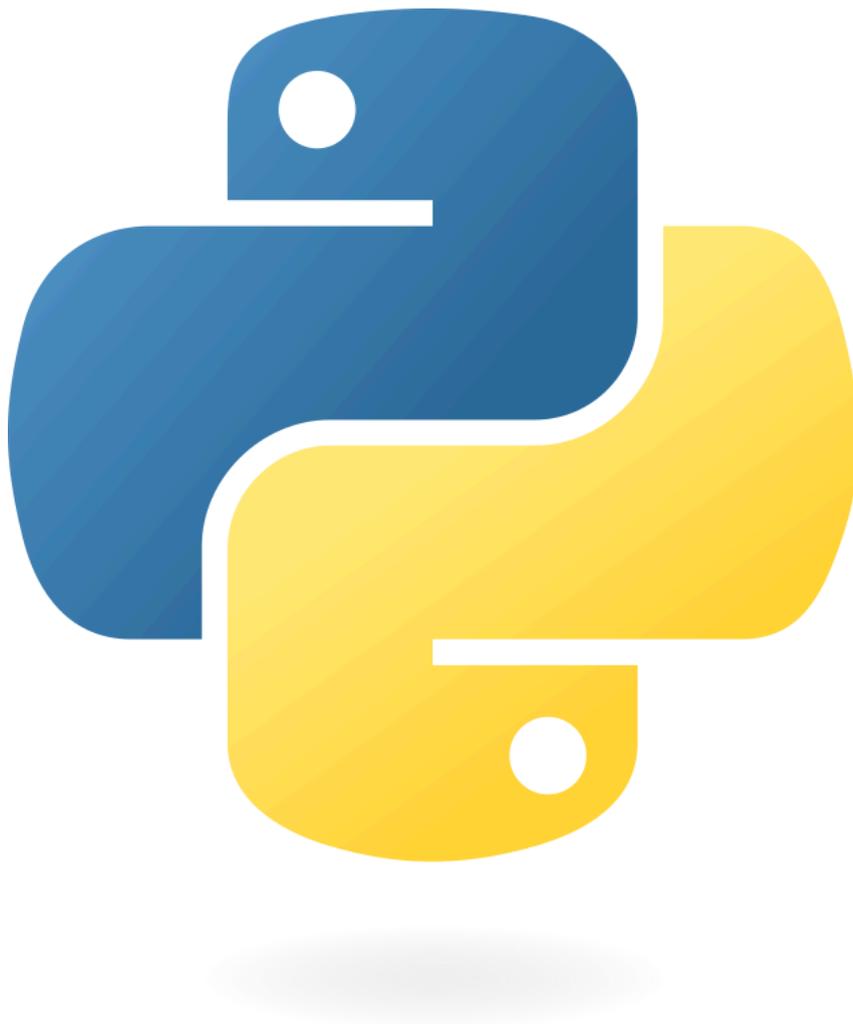


- 

### **JavaScript Password Mask**

Un esempio in JavaScript di mascheramento di una password con l'aggiunta della funzionalità di copia negli appunti.

Docker Docker Compose JavaScript



- 

### **Python Placeholder Image**

Applicazione sviluppata in Python con Flask per la creazione di immagini segnaposto.  
Docker Docker Compose Python Flask JavaScript



- 

### **Go Placeholder Image**

Applicazione in Go per la creazione di immagini segnaposto.

Docker Docker Compose Go JavaScript