

हुक्स

“हुक्स को कैसे काम चलाया जाए” का अन्वेषण करने के बाद, मैंने हुक्स के साथ हुक प्रदान की गहराई में जाकर अध्ययन किया। यहां भविष्य के संदर्भ के लिए एक संक्षिप्त अवलोकन दिया गया है।

हुक में, हुक आपको शेल ऑपरेशन के दौरान विशिष्ट बिंदुओं पर कस्टम फ़ंक्शन निष्पादित करने की अनुमति देते हैं। `preexec` के अलावा, हुक आपके वातावरण को बेहतर बनाने के लिए कई हुक प्रदान करता है:

1. `precmd`

- कब: प्रॉम्प्ट दिखाए जाने से पहले।
- उपयोग: प्रॉम्प्ट को अपडेट करें या सफाई करें।
- उदाहरण:

```
precmd() {  
    echo "      !" ;  
}
```

2. `chpwd`

- कब: जब वर्तमान डायरेक्टरी बदलती है।
- उपयोग: पर्यावरण चर को अपडेट करें या डायरेक्टरी के आधार पर क्रियाएं ट्रिगर करें।
- उदाहरण:

```
chpwd() {  
    echo "      : $PWD"  
}
```

3. `preexec_functions` और `precmd_functions`

- कब: `preexec` और `precmd` के समान, लेकिन कई फ़ंक्शन्स का समर्थन करते हैं।
- उपयोग: मौजूदा हुक्स को ओवरराइड किए बिना कई क्रियाएँ जोड़ें।

□ उदाहरण:

```
precmd_functions+=(additional_precmd)

additional_precmd() {
    echo "Additional precmd task."
}
```

4. TRAPDEBUG

□ कब: प्रत्येक कमांड के बाद, परिणाम प्रदर्शित होने से पहले।

□ उपयोग: डिबगिंग, कमांड्स को लॉग करना।

□ उदाहरण:

```
TRAPDEBUG() {
    echo "      : $1"
}
```

5. TRAPEXIT

□ कब: जब शेल बंद होता है।

□ उपयोग: सफाई कार्य या बाहर निकलने के संदेश प्रदर्शित करना।

□ उदाहरण:

```
TRAPEXIT() {
    echo "  !"
}
```

6. zle हुक्स

□ कब: लाइन संपादन के दौरान।

□ उपयोग: कमांड लाइन व्यवहार को अनुकूलित करें।

□ उदाहरण:

```

zle-line-init() {
    echo "          "
}
zle -N zle-line-init

```

7. इतिहास हुक (zshaddhistory, zshremovehistory)

- कब: इतिहास प्रविष्टियों को जोड़ने या हटाने पर।
- उपयोग: इतिहास को फ़िल्टर या प्रबंधित करने के लिए।
- उदाहरण:

```

zshaddhistory() {
    [[ $1 == *"secret"* ]] && return 1
    return 0
}

```

8. periodic

- कब: period द्वारा निर्धारित अंतराल पर।
- उपयोग: नियमित जांच या अपडेट के लिए।
- उदाहरण:

```

periodic() {
    echo "          ..."
}

```

9. add-zsh-hook

- उद्देश्य: हुक में फ़ंक्शन को सुरक्षित रूप से जोड़ना।
- उपयोग: बिना ओवरराइट किए कई फ़ंक्शन जोड़ें।
- उदाहरण:

```

add-zsh-hook precmd another_precmd

```

```

another_precmd() {

```

```
    echo "Another precmd function."  
}
```

सारांश

zsh का हुक सिस्टम बहुमुखी है, जो स्वचालन और अनुकूलन को सक्षम बनाता है:

- preexec: कमांड निष्पादन से पहले।
- precmd: प्रॉम्प्ट प्रदर्शन से पहले।
- chpwd: डायरेक्टरी बदलने पर।
- TRAPDEBUG: डिबगिंग के लिए कमांड के बाद।
- TRAPEXIT: शेल से बाहर निकलने पर।
- zle हुक्स: लाइन संपादन के दौरान।
- इतिहास हुक्स: कमांड इतिहास प्रबंधन।
- periodic: निर्धारित अंतराल पर।
- add-zsh-hook: एकाधिक हुक फ़ंक्शन जोड़ना।

इन हुक्स का उपयोग करके आप अपने zsh अनुभव को काफी बेहतर बना सकते हैं, जिससे आपका शेल अधिक कुशल और आपके वर्कफ़्लो के अनुरूप हो जाएगा।