

# String and Data Encoding

# PDF can encode data in multiple ways

```
5 0 obj
<< /Length 48 >>
stream
Hello World
endstream
endobj
```

# Hex encoding

Hello World! → Hex encoding → #48#65#6C#6C#6F#20#57#6F#72#6C#64#21

```
graph LR; A[Hello World!] --> B[Hex encoding]; B --> C[#48#65#6C#6C#6F#20#57#6F#72#6C#64#21]
```

# PDF can encode data in multiple ways

```
5 0 obj
<< /Length 48 >>
stream
Hello World
endstream
endobj
```

PDF can encode data in multiple ways

```
5 0 obj
<< /Length 48 >>
stream
#48#65#6C#6C#6F#20#57#6F#72#6C#64#21
endstream
endobj
```

# Hex, Octal, Mix and White Space Obfuscation

Hello World! → **Hex encoding** → #48#65#6C#6C#6F#20#57#6F#72#6C#64#21

Hello World! → **Octal encoding** → \110\145\154\154\157\040\127\157\162\154\144\041

Hello World! → **Mix it up** → #48\145#6C\154#6F \127#6F#72#6Cd!

Hello World! → **Add whitespace** → #48 \145#6C \154 #6F#20  
\127 #6F #72  
#6Cd !

# Using Filters to decode encoded data

```
5 0 obj
<< /Length 60 /Filter /ASCIIHexDecode >>
stream
48656c6c6f20576f726c6421111111111111
endstream
endobj
```

# Using Multiple Encoding

Filters are decoded in reverse



```
5 0 obj
<< /Length 60 /Filter [/ASCIIHexDecode/LZWDecode] >>
stream
J..)6T`?p&<!J9%_[umg"B7Z
endstream
endobj
```



# Others

**/ASCIHexDecode**

Hex encoding of  
characters

**/LZWDecode**

LZW compression  
algorithm

**/FlateDecode**

Zlib compression

**/ASCII85Decode**

ASCII base-85  
representation

**/Crypt**

Various encryption  
algorithms

# Sample malicious pdf document

```
1 0 obj
<<
  /Type /Catalog
  /OpenAction 7 0 R
>>
endobj

7 0 obj
<< /S/JavaScript/JS 8 0 R >>
endobj

8 0 obj
<< /Filter[/FlateDecode/ASCIIHexDecode]/Length 100 >>
stream
7f454c460201010000000000000000000000000000002003e0001000000000048400
00000000040000000000000000...
endstream
endobj
```