```
> monster <- c(TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE)
> print(monster)
 [1]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
> yugioh <- c("Dark Magician", "Blue Eyes", "Exodia", "Red Eyes")
> typeof(yugioh)
[1] "character"
> combined1 <- c(monster, yugioh)
> print(combined1)
 [1] "TRUE"          "TRUE"          "TRUE"          "FALSE"         "TRUE"
[12] "Blue Eyes"     "Exodia"        "Red Eyes"
> typeof(combined1)
[1] "character"
> atk <- c(2500, 3000, 9999, 2400)
> coerce.check <- c(atk, monster)
> print(coerce.check)
 [1] 2500 3000 9999 2400    1    1    1    0    1    1    1    1    1    1
> typeof(coerce.check)
[1] "double"
> as.character(monster)
 [1] "TRUE"  "TRUE"  "TRUE"  "FALSE" "TRUE"  "TRUE"  "TRUE"  "TRUE"  "TRUE"  "TRUE"
> as.numeric(monster)
 [1] 1 1 1 0 1 1 1 1 1 1
> as.logical(atk)
[1] TRUE TRUE TRUE TRUE
> as.integer(yugioh)
[1] NA NA NA NA
Warning message:
NAs introduced by coercion
> card_types <- c("Spell", "Trap", "Monster", "Spell", "Trap", "Monster")
> card_factor <- factor(card_types)
> print(card_factor)
[1] Spell   Trap    Monster Spell   Trap    Monster
Levels: Monster Spell Trap
> levels(card_factor)
[1] "Monster" "Spell"   "Trap"
> as.numeric(card_factor)
[1] 2 3 1 2 3 1
> onehot <- model.matrix(~ card_factor - 1)
> print(onehot)
  card_factorMonster card_factorSpell card_factorTrap
1                  0                1               0
2                  0                0               1
3                  1                0               0
4                  0                1               0
5                  0                0               1
6                  1                0               0
attr(,"assign")
[1] 1 1 1
attr(,"contrasts")
attr(,"contrasts")$card_factor
[1] "contr.treatment"

"TRUE"          "TRUE"          "TRUE"          "TRUE"          "TRUE"          "Dark Magician"
```

```python
from sklearn.preprocessing import LabelEncoder

colors = ["Red", "Green", "Blue", "Green", "Red"]
label_encoder = LabelEncoder()
labels = label_encoder.fit_transform(colors)
print(labels)
```

```
[2 1 0 1 2]
```

```python
import pandas as pd

df = pd.DataFrame({'Color': ["Red", "Green", "Blue", "Green", "Red"]})
onehot = pd.get_dummies(df, columns=['Color'])
print(onehot)
```

```
   Color_Blue  Color_Green  Color_Red
0       False        False       True
1       False         True      False
2        True        False      False
3       False         True      False
4       False        False       True
```

```
> colors <- factor(c("Red", "Green", "Blue", "Green", "Red"))
> print(colors)
[1] Red   Green Blue  Green Red
Levels: Blue Green Red
> levels(colors)
[1] "Blue"  "Green" "Red"
> as.numeric(colors)
[1] 3 2 1 2 3
```