test333

A fun and educational cheat sheet linking Pokemon and their pictures

Basic Types & Pokemon

Integer (Type I) - Normal

Like a Normal Pokemon, Integer (`TYPE I`) is your everyday, basic data type.

Pokemon Analogy: Rattata - common, straightforward.

Declaration:

```
DATA: lv_integer TYPE i.
```

Use:
Basic counting and numerical operations.

Example:

```
lv_integer = 5 + 3.
WRITE: lv_integer. "Output: 8
```

Character (Type C) - Fire

Character (`TYPE C`) is fiery and can contain any alphanumeric character, just like a Fire Pokemon.

Pokemon Analogy: Charmander - full of character.

Declaration:

```
DATA: lv_character TYPE c LENGTH 10.
```

Use:
Storing text strings.

Example:

```
lv_character = 'ABAP Fire'.
WRITE: lv_character. "Output: ABAP Fire
```

Numeric (Type N) - Water

Numeric (`TYPE N`) handles numerical characters, like a Water Pokemon flowing with numbers.

Pokemon Analogy: Squirtle - numeric and precise.

Declaration:

```
DATA: lv_numeric TYPE n LENGTH 8
DECIMALS 2.
```

Use:
Storing numbers as characters; good for formatted output.

Example:

```
lv_numeric = '12345.67'.
WRITE: lv_numeric. "Output: 12345.67
```

More Data Types & Pokemon

Date (Type D) - Grass

Date (`TYPE D`) represents dates, growing like Grass Pokemon.

Pokemon Analogy: Bulbasaur - evolving with time.

Declaration:

```
DATA: lv_date TYPE d.
```

Use:
Storing and manipulating dates.

Example:

```
lv_date = sy-datum.
WRITE: lv_date. "Output: Current Date
```

Time (Type T) - Electric

Time (`TYPE T`) represents time, quick and electric like an Electric Pokemon.

Pokemon Analogy: Pikachu - fast and zappy.

Declaration:

```
DATA: lv_time TYPE t.
```

Use:
Storing and manipulating time.

Example:

```
lv_time = sy-uzeit.
WRITE: lv_time. "Output: Current Time
```

String - Ghost

String is dynamic and flexible, like a Ghost Pokemon.

Pokemon Analogy: Gastly - Ethereal, adaptable.

Declaration:

```
DATA: lv_string TYPE string.
```

Use:
Storing variable-length character strings.

Example:

```
lv_string = 'ABAP is cool'.
WRITE: lv_string. "Output: ABAP is cool
```

ABAP Objects & Pokemon Evolution

Classes - Legendary Pokemon

ABAP Classes are like Legendary Pokemon - rare, powerful, and defining.

Pokemon Analogy: Mewtwo - strong, unique, and complex.

Definition:

```
CLASS lcl_legendary DEFINITION.
  PUBLIC SECTION.
    METHODS: power_up.
ENDCLASS.
```

Implementation:

```
CLASS lcl_legendary IMPLEMENTATION.
  METHOD power_up.
    WRITE: 'Legendary Power!'.
  ENDMETHOD.
ENDCLASS.
```

Usage:

```
DATA: lo_legendary TYPE REF TO lcl_legendary.
CREATE OBJECT lo_legendary.
lo_legendary->power_up(). "Output: Legendary Power!
```

Interfaces - Eevee's Evolution

Interfaces are like Eevee - they can evolve into different forms and provide versatility.

Pokemon Analogy: Eevee - adaptable to many types.

Definition:

```
INTERFACE lif_evolution.
  METHODS: special_attack.
ENDINTERFACE.
```

Implementation:

```
CLASS lcl_fire IMPLEMENTATION.
  METHOD special_attack.
    WRITE: 'Fire Attack!'.
  ENDMETHOD.
ENDCLASS.
```

Usage:

```
DATA: lo_evolution TYPE REF TO lif_evolution.
CREATE OBJECT lo_fire.
lo_evolution = lo_fire.
lo_evolution->special_attack(). "Output: Fire Attack!
```

ABAP Statements & Pokemon Moves

SELECT Statement - Pokemon Catching

The `SELECT` statement is like catching a Pokemon - fetching data from the database.

Pokemon Analogy: Poke Ball - capturing the right data.

Syntax:

```
SELECT * FROM pokemon_table INTO TABLE
@DATA(lt_pokemon).
```

Use:
Retrieving data from a database table.

Example:

```
LOOP AT lt_pokemon ASSIGNING FIELD-
SYMBOL(<fs_pokemon>).
  WRITE: / <fs_pokemon>-name.
ENDLOOP.
```

WRITE Statement - Pokemon Display

The `WRITE` statement is like showing off your Pokemon - displaying the data.

Pokemon Analogy: Pokedex - displaying information.

Syntax:

```
WRITE: lv_pokemon_name.
```

Use:
Displaying data on the screen.

Example:

```
WRITE: / 'Pokemon Name:',
lv_pokemon_name.
```

LOOP AT Statement - Pokemon Training

The `LOOP AT` statement is like training Pokemon - processing each entry in a table.

Pokemon Analogy: Training Session - working through each Pokemon.

Syntax:

```
LOOP AT lt_pokemon ASSIGNING FIELD-
SYMBOL(<fs_pokemon>).
  "Process each pokemon here.
ENDLOOP.
```

Use:
Iterating through internal tables.

Example:

```
LOOP AT lt_pokemon ASSIGNING FIELD-
SYMBOL(<fs_pokemon>).
  <fs_pokemon>-level = <fs_pokemon>-
level + 1.
ENDLOOP.
```