I can't directly create or share PDFs, but I can help you compile all the information into a structured document that you can easily save as a PDF. Here's how we'll do it:

Complete Guide to JavaScript Object Iteration & Filtering

(All Previous Explanations Combined & Organized)

---

1. The Core Concept

What this code does:

```javascript
for (const city in citiesPopulation) {
  if (city === "Berlin") break;
  cityNewPopulations[city] = citiesPopulation[city];
}
```

- Loops through city names in `citiesPopulation`

- Copies populations to a new object (`cityNewPopulations`)

- Stops when it reaches "Berlin"

---

2. Key Components Explained

| Part | What It Does | Why It Matters |

|------|-------------|----------------|

| `for...in` | Iterates over object keys (city names) | Accesses each city one by one |

| `city` | Temporary variable storing current key | Holds names like "London", "New York" |

| `citiesPopulation[city]` | Gets population for current city | Retrieves values (e.g., 8900000 for "London") |

| `cityNewPopulations[city] = ...` | Copies to new object | Creates filtered dataset |

---

3. Step-by-Step Execution

Original Data:

```javascript
let citiesPopulation = {
  "London": 8900000,
  "New York": 8400000,
  "Berlin": 3500000
};
```

How the Loop Runs:

1. First pass: Copies `"London": 8900000`
2. Second pass: Copies `"New York": 8400000`
3. Sees "Berlin" → `break` → Loop ends

Final Output:

```javascript
{
  "London": 8900000,
  "New York": 8400000
}
// Berlin is excluded!
```

---

4. Visual Analogy

Like sorting mail:

- Original box (`citiesPopulation`) has all letters

- New box (`cityNewPopulations`) starts empty

- You copy letters until you find one marked "Berlin" → stop

---

5. Why This Technique Matters

- **Preserves original data** → Safe for analysis

- **Memory efficient** → Only copies what you need

- **Foundation for:**

  - Data filtering

  - Transformations

  - Clean data exports

---

6. Pro Tips

1. **For arrays:** Use `for...of` instead

2. **Deep copies:** For nested objects, use:

   ```javascript
   JSON.parse(JSON.stringify(object))
   ```

3. **Modern alternative:**

   ```javascript
   ```

```
  Object.entries(citiesPopulation).forEach(([city, pop]) => {
    if (city !== "Berlin") cityNewPopulations[city] = pop;
  });
```

---

7. Common Pitfalls

Mistake	Fix
Including prototype properties	Add `if (object.hasOwnProperty(key))`
Modifying original object	Always copy to new object first
Assuming order matters	Use `Map` if order is critical

How to Save as PDF

1. **Copy this entire formatted text**
2. **Paste into a Word/Google Docs document**
3. **Export as PDF** (File > Download > PDF)

Would you like me to:

- Add more practical examples?
- Include a comparison with other looping methods?
- Provide exercises to test your understanding?

Let me know how I can enhance this for you! 🚀